

Proyecto de Grado en Informática.

Sistema de Recomendación con conocimiento Web



MEMORIA DEL PROYECTO

Eddy Cuizaguana Cerpa

Leandro Roberto Yáñez Torres

Dirigido por: Juan A. Recio García

Departamento de Ingeniería del Software e Inteligencia Artificial

Facultad de Informática

Universidad Complutense de Madrid

Junio 2015

Sistema de Recomendación con conocimiento Web: Un Sistema de Recomendación para Hoteles con conocimiento web

Memoria de Proyecto de Sistemas Informáticos

Ingeniería del Software e Inteligencia Artificial

06/2015

Departamento de Ingeniería del Software e Inteligencia

Artificial

Facultad de Informática

Universidad Complutense de Madrid

Junio 2015

Autorización

Eddy Cuizaguana Cerpa y Leandro Roberto Yáñez Torres, alumnos matriculados en la asignatura de Trabajo final de grado (TFG), autorizan, mediante el presente documento, a la Universidad Complutense de Madrid (UCM), a difundir y utilizar con fines académicos, no comerciales, y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado. Todo ello realizado durante el curso académico 2014-2015 bajo la dirección de Juan Antonio Recio García, profesor del Departamento de Ingeniería del Software e Inteligencia Artificial.

Eddy Cuizaguana Cerpa

Leandro Roberto Yáñez Torres

*Cuando no te das cuenta de lo que no puedes hacer,
puedes hacer cosas muy curiosas.*

Agradecimientos

Para nosotros esta página representa el *Main* de nuestro proyecto, importante sobre todo para reconocer con gratitud a todas las personas que con gran capacidad humana y un talento demostrado, nos han dado la mano, una guía, apoyo y su ayuda para lograr llegar hasta este punto de nuestras carreras y vidas.

Agradecemos a Juan Antonio Recio García en primer lugar por darnos la confianza de ser parte de éste proyecto y trabajar bajo su dirección para desarrollar nuestro trabajo de fin de carrera, estamos agradecidos porque hemos sido conscientes del tiempo y dedicación que nos ha otorgado en las reuniones ya sea para presentaciones, correcciones y resolución de problemas, gracias porque cuando estuvimos en un bucle infinito nos ayudó a encontrar la salida y con toda esta ayuda llevar a cabo nuestros objetivos.

Muchas gracias a todos los compañeros que han sido parte de nuestra vida universitaria y que se han sentido identificados de algún modo con nuestros esfuerzos para lograr que este proyecto avance, tanto a aquellos que nos han ayudado con un par de palabras inspiradoras hasta los que se han sentado con nosotros a resolver problemas de cualquier asignatura en general, gracias por su ayuda y amistad (Mariano, Jorge, Samuel, Stephania, Laura, Omar y Manuel).

A quienes más podríamos dar las gracias sino a nuestras queridas familias, que con una gran capacidad de amor nos han llenado de sueños e ilusiones la vida, justamente ahora estamos a un paso de cumplir uno de esos sueños compartidos, todo gracias a su incondicional ayuda. Los amamos y estamos infinitamente agradecidos por tanto amor, comprensión, cariño, paciencia y un sinfín de cosas que han sido necesarias para lograr esta meta, sin su apoyo no estaríamos aquí.

A todos ustedes gracias totales!!

Les dedicamos con cariño y admiración todo el trabajo que nos representó hacer este proyecto.

Resumen

Una parte importante de la web que hoy manejamos cuenta con la presencia de sistemas de recomendación y éstos se van apoderando cada vez más de las aplicaciones más comunes y de mayor consumo por los usuarios, entre estas aplicaciones se sitúan los buscadores, aplicaciones de compra y en cualquier sitio donde se maneje la toma de decisión de un usuario o se muestran opciones de acuerdo a las necesidades que presenten en ciertos campos los clientes, con menor o mayor refinamiento tendremos un sistema de recomendaciones funcionando muchas veces de manera inadvertida para mucha gente.

El crecimiento es notorio y la complejidad de estos sistemas ha crecido, son mucho más potentes, pueden por ejemplo casi adivinar qué podrías comer un miércoles al mediodía, o si la película de estreno que está en las salas de cine es de tu preferencia y hasta planificar tus vacaciones, es justamente en este último caso donde enfocamos nuestro proyecto, aprovechando el mercado de turismo en España que es ricamente productivo el cual ha generado mucho beneficio a lo largo del tiempo, en conocimiento que España ha sido, es y será un destino turístico por excelencia, tener al alcance herramientas que faciliten al usuario la toma de decisiones en cuanto a qué lugar hospedarse siendo el mismo el que más se ajuste a sus gustos y necesidades es primordial, en la web hay varios sistemas que ya realizan este tipo de servicios como Tripadvisor o Trivago entre otros, nosotros hemos creado el nuestro, dedicado al sector hotelero para las principales y más visitadas ciudades de España.

Hemos desarrollado bajo la perspectiva mencionada antes, un Sistema Recomendador de Hoteles con conocimiento Web, bajo dos conceptos que hemos llamado por sus características, a uno analizador estadístico y a otro analizador sentimental, la idea general es que un usuario pueda evaluar un hotel de su interés de varias formas en nuestra aplicación, de las que entraremos en detalle más adelante en esta memoria, saber la reputación de dicho hotel bajo el análisis estadístico y también con un análisis sentimental que obedece a manejo de la información web que se encuentre relacionada al hotel en cuestión, con el fin de dar una evaluación en cada caso, mostrando una puntuación, dicho resultado llevará al usuario a tomar la decisión más adecuada que se perfile a sus necesidades, siendo esta última parte el objetivo principal que tenemos con nuestro proyecto, que es satisfacer de forma eficiente las dudas y preferencias de los usuarios, sin que pierdan el control de decidir y tener siempre la última palabra.

TripAdvisor, Trivago, Sistema Recomendador, Google, API, Hoteles, Análisis Sentimental, Java, valoración

Abstract

An important part of the web that we now handle counts on the presence of recommendation systems which are taking over more and more of the most common applications and most demanded by users, among these we find the application's searchers, applications where there's a purchase, and anywhere where the user makes a decision or options are shown according to the customers' needs that may arise in certain fields, with more or less refinement but we'll have a recommendation system running often inadvertently for many people.

Growth is notorious and the complexity in these systems has grown, they're so powerful that can guess, for example, what you could eat on a Wednesday at noon, or if the movie that's in theatres in premiere is of your choice and even plan your vacations, it's precisely in this last one that we focus our project, taking advantage of the fact that the tourism market in Spain is richly productive which has generated much interest over time, knowing that Spain has been, is and will be a tourist destination, having tools available to facilitate user's decisions as to which place to stay at that is the one that best suits the user's tastes and needs is essential, on the web there's several systems that already offer this kind of services like Tripadvisor or Trivago among others, we have created our own, dedicated to the hotel sector for the main and most visited cities in Spain.

We have developed under the perspective mentioned earlier, a Recommender Hotels System under two concepts which we call, by its nature, one of statistical analyser and other of emotional analyser, the general idea is that an user can evaluate a hotel of their choice in several ways through our application which will go into detail later in this report, know the reputation of that same hotel under the statistical analysis and also with the sentimental analysis that obeys to the web information management that is found related to the concerned hotel, in order to make an assessment in each case, showing a score, which will take the user to make the right decision that outlines their needs, being that last part the main objective we have with our project, which is to satisfy efficiently the user's doubts and preferences, without losing the control of decision making and always having the last word.

TripAdvisor, Trivago, Recommender Systems, Google, API, Hotels, Sentimental analysis, Java, valuation

Índice

1	Introducción.....	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Estructura de la memoria.....	3
	Introduction	5
	Motivation.....	5
	Objetives	6
	Report's Structure	7
2	Estado del arte	9
2.1	Sistemas de recomendación	9
2.1.1	Tipos de sistemas de recomendación	10
2.2	Investigación web.....	11
2.2.1	Sistema de Recomendación TripAdvisor-Trivago.....	11
2.2.2	Keywords y expresiones regulares	12
2.2.2.1	Keywords y expresiones regulares positivas.....	14
2.2.2.2	Keywords y expresiones regulares negativas.....	15
2.3	Información del contexto y del usuario	17
3	Explicación Teórica	19
3.1	Sistema de Recomendación de Hoteles	19
3.1.1	Analizadores	21
3.2	Características Recomendador de Hoteles	25
3.3	Perfil del usuario	25
3.4	Categorización de keywords y expresiones regulares	26
3.5	Sistema de valoración	27
3.6	Algoritmos	28
3.6.1	Estadístico vs Sentimental.....	28
3.6.2	Búsqueda Sentimental	33
3.7	Conclusiones.....	35
4	Implementación	37
4.1	Arquitectura Backend.....	37
4.1.1	Funcionalidad y diseño.....	37

4.1.2	Tecnologías.....	39
4.1.3	Tecnologías utilizadas.....	39
4.1.3.1	Java.....	40
4.1.3.2	Jersey.....	40
4.1.3.3	JSON	41
4.1.3.4	Gson.....	41
4.1.4	Diseño UML Backend	41
4.1.5	API (Interfaz de programación de aplicaciones)	45
4.2	Arquitectura Frontend	49
4.2.1	Tecnologías usadas para la implementación del Frontend.....	50
4.2.1.1	HTML5	50
4.2.1.2	CSS3	51
4.2.1.3	JavaScript.....	51
4.2.1.4	JQuery.....	51
4.2.1.5	Bootstrap.....	52
4.2.1.6	Material	52
4.2.2	Diseño UML Frontend	52
5	Descripción Funcional.....	59
5.1	Definir posición	59
5.2	Buscar hoteles cercanos.....	61
5.3	Analizador Estadístico y Sentimental	63
6	Test	69
6.1	Conexión.....	69
6.2	Resultados	72
6.2.1	Estudio del análisis estadístico	72
6.2.2	Estudio del análisis sentimental	75
6.2.3	Estadístico vs Sentimental.....	77
7	Conclusiones y Trabajo Futuro	81
7.1	Conclusiones.....	81
7.2	Trabajo Futuro.....	82
	Conclusions and Future Work	83
	Conclusions	83
	Future Work	84
8	Apéndice A.....	85

8.1	Contribuciones y trabajo individual	85
8.1.1	Eddy Cuizaguana Cerpa	85
8.1.2	Roberto Yáñez Torres.....	88
9	Bibliografía	93

Índice de figuras

Figura 2.1 Comentario de un usuario usando palabras positivas	15
Figura 2.2 Comentario de un usuario usando palabras negativas	17
Figura 3.1 Captura de pantalla del analizador estadístico	22
Figura 3.2 Captura de pantalla del analizador sentimental	23
Figura 3.3 Captura de pantalla de los resultados del analizador sentimental.....	23
Figura 3.4 Esquema del servicio ofertado a Madrid Manía	24
Figura 3.5 Captura de pantalla de nuestro sistema de valoración.....	28
Figura 3.6 Captura de pantalla de una búsqueda avanzada en Google.....	29
Figura 3.7 Diagrama de estados del algoritmo estadístico	31
Figura 3.8 Captura de pantalla del resultado estadístico obtenido	33
Figura 3.9 Diagrama de estados del algoritmo sentimental	35
Figura 4.1 Arquitectura backend.....	39
Figura 4.2 Diagrama con las clases utilizadas para realizar la conexión con el servidor	42
Figura 4.3 Diagrama que representa los hoteles cercanos a una posición.....	42
Figura 4.4 Clases (Precio, Servicio y Ubicación) que implementan nuestro diccionario	43
Figura 4.5 Diagrama que representa nuestro diccionario de palabras positivas y negativas.....	43
Figura 4.6 Diagrama con las clases auxiliares utilizadas	44
Figura 4.7 Diagrama con clases utilizadas para implementar el análisis estadístico	44
Figura 4.8 Diagrama con clases utilizadas para implementar el análisis sentimental	45
Figura 4.9 Arquitectura Frontend.....	50
Figura 4.10 Diagrama de secuencia UML para obtener la ubicación en el mapa	53
Figura 4.11 Diagrama de secuencia UML de la petición de hoteles cercanos	54
Figura 4.12 Diagrama de secuencia UML de la petición del análisis estadístico	55
Figura 5.1 Definir posición ciudad	60
Figura 5.2 Definir posición mapa y geolocalización	61
Figura 5.3 Hoteles cercanos	62
Figura 5.4 Selección de hotel	63
Figura 5.5 Analizador estadístico y sentimental	64
Figura 5.6 Resultado analizador estadístico.....	65
Figura 5.7 Animación de espera análisis sentimental	65

Figura 5.8 Resultado análisis sentimental.....	66
Figura 6.1 Conexión con TOR	69
Figura 6.2 Erro 503	69
Figura 6.3 Redirección de Google	70
Figura 6.4 Aplicación UltraSurf.....	71
Figura 6.5 Conexión a un proxy.....	71
Figura 6.6 Método que obtiene el HTML de Google.....	71
Figura 6.7 Grafica MAE estadístico por ciudades.....	73
Figura 6.8 Gráfica MAE estadístico según categoría (estrellas)	74
Figura 6.9 Gráfica MAE sentimental por ciudades.....	76
Figura 6.10 Gráfica MAE sentimental según categoría (estrellas)	77
Figura 6.11 Gráfica MAE estadístico vs sentimental por ciudades	79
Figura 6.12 Gráfica MAE estadístico vs Sentimental según categoría (estrellas)	80
Figura 8.1 Teorema de Bayes	89
Figura 8.2 Google Search Box.....	90
Figura 8.3 Google Search Places.....	91

Índice de tablas

Tabla 2.1 Palabras claves.....	14
Tabla 3.1 Ejemplos de subcategorías de Servicios y sus términos.....	26
Tabla 3.2 Ejemplos de subcategorías de Precio y sus términos.....	26
Tabla 3.3 Ejemplos de subcategorías de Ubicación y sus términos	27
Tabla 4.1 JSON con los hoteles cercanos	46
Tabla 4.2 JSON con los resultados webs evaluados	47
Tabla 4.3 JSON con el resultado estadístico obtenido	48
Tabla 4.4 JSON con el mejor hotel	49
Tabla 4.5 Descripción de la ubicación en el mapa	54
Tabla 4.6 Descripción de la petición de hoteles.....	55
Tabla 4.7 Descripción de la petición del análisis estadístico.....	56
Tabla 4.8 Diagrama de secuencia UML del análisis sentimental	56
Tabla 4.9 Descripción de la secuencia del análisis sentimental	57
Tabla 5.1 Definir posición ciudad	60
Tabla 5.2 Definir posición mapa y geolocalización	61
Tabla 5.3 Hoteles cercanos	62
Tabla 5.4 Selección de hotel.....	63
Tabla 5.5 Analizador estadístico y sentimental.....	64
Tabla 5.6 Selección web del resultados sentimental	66
Tabla 5.7 Resultado detallado del análisis sentimental.....	67
Tabla 6.1 Resultado MAE estadístico por ciudades	73
Tabla 6.2 Resultado MAE estadístico según categoría (estrellas).....	74
Tabla 6.3 Resultado MAE sentimental por ciudades	76
Tabla 6.4 Resultado MAE sentimental según categoría (estrellas).....	77
Tabla 6.5 MAE estadístico vs Sentimental por ciudades.....	78
Tabla 6.6 MAE estadístico vs Sentimental según categoría (estrellas)	80

1 Introducción

En esta parte de introducción hablaremos de lo que nos ha motivado a realizar el proyecto Sistema Recomendador de Hoteles con conocimiento Web SRH, su inicialización y objetivos, así como también el desarrollo del mismo, finalmente detallaremos la estructura general de lo que se verá en esta memoria.

1.1 Motivación

España uno de los países más turísticos del mundo es visitado al año por unos sesenta y picos millones de turistas (ABC, 2014)provenientes de todas partes del mundo, quienes necesitan en determinado momento un sitio para alojarse, las de pernoctaciones en establecimientos españoles ha crecido mucho en los últimos años y tiende a ser mejor, lo dice así los informes del Instituto Nacional de Estadística INE_(INE, 2015)y por el afán de encontrar un lugar adecuado a sus necesidades los mismos realizan determinadas búsquedas utilizando varias aplicaciones o visitando muchas opciones de buscadores en la web para hallar el lugar que cumpla todas o muchas de sus demandas.

Tripadvisor o Trivago pueden ser buenos ejemplos de los que hablamos, pues dadas ciertas pautas por parte de los usuarios que son en realidad las características que buscan o que quieren que un hotel cumpla y van personalizando su búsqueda con ciertos filtros que estas aplicaciones prestan, nuestro Sistema de Recomendación de Hoteles se basa en estos servicios, queremos dar al usuario opciones para encontrar hoteles que realmente encuentre atractivos.

Los sistemas de recomendación usan información dinámica del usuario así como su posición en alguna parte geográfica y datos relevantes a su información personal, uso de historiales de búsqueda web, cookies, etc. Para aproximar positivamente los resultados que se obtengan en una recomendación, SRH respeta esta información personal de cada usuario, usa dos métodos de recomendación: **estadístico** y **sentimental**, algo que puede contrastar los resultados e influenciar en una decisión positiva para el usuario, ayudándolo a que tienda a la mejor opción, usando por una parte resultados numéricos de las páginas que hablan de cierto hotel y por otro un procesamiento de lenguaje de los comentarios de otras personas que ya han pasado por la experiencia de alojarse en un hotel que puede ser candidato a ser elegido por el usuario.

Que una persona de cualquier parte del mundo quiera visitar España y no tenga complicaciones con lo que representaría buscar un hotel adecuado es nuestra motivación, dando una reputación o valoración a cada búsqueda de un hotel específico que se realice, a partir de la cual el usuario obtenga información comprensible y pueda elegir entre las recomendaciones realizadas.

1.2 Objetivos

El objetivo de este proyecto es dar un servicio web para turistas que visiten las principales ciudades de España con diseño que se adapta a todos los dispositivos, pues el servicio puede funcionar desde un computador personal hasta dispositivos móviles, brindando al usuario un Sistema Recomendador de Hoteles con conocimiento Web avanzado que obtendrá información de las webs tratándola estadísticamente y también en un análisis sentimental procesando el lenguaje natural, reflejando valoraciones para cada hotel seleccionado por el usuario dando como resultado la reputación virtual de un determinado hotel.

Los objetivos parciales que hemos llevados a cabo para cumplir el objetivo principal son los siguientes:

1. Integrar los servicios el API de Google y Google Maps para trabajar con los eventos del mapa y manejar información del usuario de forma dinámica, como es su localización.
2. Crear un buscador de hoteles cercanos a la posición del usuario, hasta 500 m. o simplemente a la de un punto x de una ciudad.
3. Formalizar los servicios más demandados en las búsquedas de hoteles, así como también los demás features para un análisis estadístico.
4. Crear un analizador estadístico usando features como el precio, la ubicación y servicios del hotel a analizar, siguiendo modelos bayesianos los cuales generan resultados mostrando al usuario un start ranking como recomendación.
5. Categorizar las expresiones regulares y palabras claves que encontramos en la web para su clasificación y análisis que con un procesamiento de lenguaje podamos utilizar un analizador sentimental.
6. Realizar un analizador sentimental que escaneara páginas web con el afán de procesar la información social de los usuarios que se relacionan con un hotel específico para finalmente recomendarlo usando valoraciones que representan la reputación del hotel.
7. Crear una página web como interfaz para el usuario para dar el servicio de nuestra aplicación con diseño responsive, de esta manera se pueda adaptar a las actuales demandas de movilidad de los usuarios, y sirva para interactuar con el mapa de y visualizar los resultados y valoraciones.
8. Crear módulo de adaptación para brindar servicios específicos a aplicaciones que soliciten de alguna funcionalidad de nuestro SRH, como Madrid Manía.

9. Contrastar los resultados de los dos analizadores anteriores con información real, para observar cuál de los dos métodos se aproxima más a un servicio web popularmente usado y analizar en función del MAE.

1.3 Estructura de la memoria

La estructura de esta memoria está compuesta de las siguientes partes consecutivas:

1. El primer capítulo está dedicado a la Motivación en crear un Sistema Recomendador de Hoteles con conocimiento Web para el país, el Objetivo que deseamos cumplir con nuestro proyecto y la estructura de esta memoria de trabajo.
2. En el capítulo 2 se presenta el Estado del Arte, en el cual describimos los sistemas de recomendación en general, introduciendo un concepto que explique su función básica, también veremos una clasificación de estos sistemas y su uso en diferentes campos como es el hotelero y de esta forma ir perfilando la descripción con nuestro proyecto. Hablaremos de los Sistemas de Recomendación que ya existen en el mercado a los cuales hemos analizado y tomados como referencia para desarrollar nuestra aplicación, hemos añadido la parte de Investigación Web para extendernos con los conceptos investigados y las fuentes de información que usamos a lo largo del proceso de nuestro trabajo. Por otra parte definimos las expresiones regulares y las palabras claves así como su clasificación dentro del contexto que hemos manejado para este proyecto.
3. El capítulo 3 trata de la Explicación Teórica de las funcionalidades de nuestro SRH, daremos una introducción breve de nuestra aplicación sobre su funcionamiento, hablaremos también de sus características, añadimos a esta parte de la memoria la categorización de los features para su análisis e interpretación. Además mostraremos en este capítulo el sistema de valoración detallando las métricas utilizadas para la valoración de un hotel, explicaremos de forma general los algoritmos creados para los distintos servicios de nuestra aplicación, pondremos la información del contexto ubicando el entorno en el que los usuarios manejan nuestra aplicación y finalizamos con las conclusiones de toda la información de este capítulo.
4. Se muestra en el capítulo 4 la Implementación, es decir la arquitectura Backend y Frontend, explicaremos su funcionalidad, diagramas UML, casos de uso, la información que utiliza, el procesamiento y origen dicha información, los resultados que ofrece cada arquitectura y hablaremos también de las tecnologías que fueron usadas en cada parte de nuestro proyecto muy brevemente.
5. Ya en el capítulo 5 nos preparamos para ver la Descripción Funcional, donde hablaremos del funcionamiento lógico del recomendador visto desde la parte del Frontend, para mayor organización ponemos en evidencia los pasos a seguir para

alguna actividad o tarea que el usuario demande y los casos de uso relacionados a cada actividad.

6. En el capítulo 6 damos a conocer el Test realizado con 160 hoteles como muestra, la dificultad que nos supuso realizarlo, mostramos además una serie de gráficas que nos permite tener una visualización más genérica de nuestras puntuaciones.
7. Conclusiones, un capítulo sin numerar, destinado a las deducciones de todo el proceso creativo, de implementación y análisis de resultados de nuestro proyecto de fin de carrera el Sistema Recomendador Hotelero para las principales y mayormente turísticas ciudades españolas.
8. Trabajo Futuro, hablamos aquí de las posibles extensiones de este proyecto, la escalabilidad que tiene en un futuro no muy lejano y posibles mejoras para un refinamiento más preciso en la recomendación.
9. Bibliografía haremos una revisión de las referencias que han sido más útiles y de las que inspiraron y ayudaron a nuestro trabajo, todas las referencias bibliográficas que hemos podido rescatar.
10. Apéndice A, en el hablaremos de las contribuciones en grupo e individuales, para el desarrollo del SRH, así como las funcionalidades que realizamos cada uno para complementarlas y armar nuestro proyecto de fin de carrera.

Introduction

On this part of the introduction we'll speak about what has motivated us into doing the project Hotel System Recommendation HSR, its beginning and objectives, and also its internal development, we'll describe the structure of what will be seen on this report.

Motivation

Spain is one of the most touristic countries in the world, is visited by around seventy million tourists per year (ABC, 2014) from everywhere in the world, who need at some point a place to stay, the overnights in Spanish establishments have grown considerably on the last few years and tend to improve, information provided by the National Institute of Statistics INE (INE, 2015) and because of the eagerness to find a suitable place which matches their needs, the consumers make certain searches using several applications or visiting various search engine options on the web to find the place that fulfills their most or every of their demands.

Tripadvisor or Trivago can be good examples of what we're writing about, because by using certain user's guidelines that are in reality the features that the guests are looking for in a hotel to follow, features which they personalize in their search with certain filters that these applications provide, our System Recommender is based on these services, we want to give the user options to find hotels that they find truly attractive.

The system recommenders use the user's dynamic information like its geographic location and relevant data about their personal information, use of web search histories, cookies, etc. To positively approximate the results on a recommendation, HSR respects each user's personal information, uses the methods of recommendation: **statistic** and **sentimental**, something that can contrast the results and influence a positive decision for the user, helping them decide on the better option, using on one hand numerical results of the pages that speak of a certain hotel and on the other hand, a language processing of other people's comments that have gone through the experience of staying at an hotel that may be an option for the user.

Our motivation is that a person from any part of the world that would like to visit Spain will have trouble finding a suitable hotel, giving to each user's search of a specific hotel a reputation or valuation, from where the user may get understandable information and select among the given recommendations.

Objectives

The objective of this project is to provide a web service for tourists that visit the main cities in Spain with a design that's for every devices, because the service can function from a personal computer or from mobile devices, offering to the user an advanced hotel system recommender that will obtain information from the web, analyzing it statistically and also by a sentimental analysis processing the natural language, reflecting valuations for each hotel selected by the user, giving as a result a virtual reputation of a certain hotel.

The partial objectives that we have accomplished to fulfill the main goal are:

1. To integrate the services, Google's API and Google Maps to work with the map events and manage user's information dynamically, as their location.
2. To create a nearby hotel search to the user's location, up to 500 m. or simply to a certain point of a city.
3. To formalize the most demanded services on the hotel searches, and also the remaining features to a statistical analysis.
4. To create a statistical analyzer by using features like the price, the location and hotel services, following Bayesian models that generate results, showing the user a ranking star as recommendation.
5. To categorize the regular expressions and keywords that we find on the web to classify and analyze that, with a language processor, we can use a sentimental analyze.
6. To do a sentimental analyzer that scans web pages to process the user's social information that is related to a specific hotel, to finally recommend using valuations that represent the hotel's reputation.
7. To create a web page as an interface for the user to provide our responsive design application service, adapting the current mobility demands, and to interact with the map and visualize the results and valuations.
8. To create an adaptation module to provide specific services to applications that require any feature of our SHR, like Madrid Manía.
9. To contrast the results of the previous analyzers with real information, in order to observe which one of the methods is closer to a more popular web service and analyze the MAE function.

Report's Structure

The structure of this report is divided in the following parts:

1. The first chapter is dedicated to the Motivation in creating a Hotel System Recommender for the country, the Objective which we hope to fulfill with our project, and the Structure of this Report.
2. On the second chapter we present the State of the Art, in which we describe the system recommenders in general, introducing a concept that explains its basic function, we'll also see a qualification of these systems and its use in different fields like the hotel market, profiling the description of our project. We'll speak about the Recommendation Systems that already exist on the market, of which we have done an analysis and taken reference to develop our application, we have added the part of the Web Investigation to extend with the investigated concepts and the sources of information that we have used throughout this process. On the other hand, we define the regular expressions and the keywords and also its qualification inside the context that we have worked with in this project.
3. The third chapter is about the Theoretical Explanation of the functionalities of our SHR, we will give a brief introduction about its functioning, we'll talk about its features, we'll add to this part of the report the categorization of the features to its analysis and interpretation. Besides this, we'll show on this chapter the valuation system, explaining in detail the metrics used for the valuation of a hotel, we'll do a general explanation of the algorithms created to the distinct services of our application, we'll put the information on the context, orienting the environment in which the users manage our application, and we'll finalize with the conclusions of all the information of this chapter.
4. On the fourth chapter, we demonstrate the Implementation or Backend and Frontend architecture, we'll explain its function, UML diagrams, cases of use, the information that it uses, the processing and where it gets that information, and of course the results that are offered in each architecture. We'll also speak about the technologies that were used in each part of the project very briefly.
5. On the fifth chapter, we'll prepare ourselves to see the Functional Description, where we'll talk about the logical functioning of the recommender, seen from the Frontend for bigger organization, we'll put in evidence the steps to follow to some activity or task that the user requires and the cases of use related to each activity.
6. On the sixth chapter we demonstrate the Test done with 160 hotels as a sample, the difficulty of this task, we'll also show a series of graphics that allow us to have a more generic visualization of our punctuations.

7. Conclusions, a chapter without a number, destined to reflections about all the creative process, implementation and analysis of results of our final graduation project, the Hotel System Recommender to the main touristic Spanish cities.
8. On Future Work, we speak about the possible extensions of this project, the scalability that the near future has and possible improvements to a more precise refinement on the recommendation.
9. On the Bibliography, we'll do a revision of what has been more useful and of what has inspired and helped in our project, all the bibliographical references that we have been able to gather.
10. On the Appendix A, we'll speak about the individual and group contributions, to the development of the SHR, and also the functionalities that each one of us has done to complement and strengthen our final graduation project.

2 Estado del arte

2.1 Sistemas de recomendación

Los sistemas de recomendación son herramientas y técnicas software que proporcionan sugerencias de elementos que pueden interesar en gran porcentaje a los usuarios_(Ricci, Rokach, & Shapira, 2011), éstas sugerencias entran a un proceso de toma de decisiones, que podría ser automatizado o dejar a la elección del usuario, todo dependerá del tipo de recomendador que estemos manejando y a qué sector va dirigido, cuando la situación es vender variedades de elementos como música, películas, ropa, tecnología, etc. se nota cierto refinamiento que tiene que ver con la parte comercial y de marketing. Los sistemas recomendadores SR son sistemas inteligentes que obteniendo cierta información son capaces de “adivinar” los gustos de una persona.

Se puede definir también a un sistema recomendador como un buscador avanzado, es decir aquel buscador que nos genera resultados los más acertados posibles, todos ellos podrían servirle al usuario y éste darlos como buenos a todos, pues si lo vemos de esta forma el SR sería un buscador bastante potente con filtros inteligentes incorporados y éstos relacionados a la información de aficiones o gustos de un usuario quien hace la búsqueda de algún elemento determinado_(Wikipedia, 2015).

¿Qué beneficios puede traer el uso de un sistema de recomendaciones?, vamos a mostrar algunos ejemplos que podemos usar:

- Uno de los más importantes es incrementar la satisfacción del usuario, tendremos a un usuario contento si obtiene resultados que realmente esperaba o incluso que superen sus expectativas, agradecerá sugerencias inteligentes que sean verdaderas soluciones a sus problemas.
- Para sistemas de recomendación comerciales aumentar el número de artículos vendidos es el objetivo, pues vender elementos adicionales a los que normalmente se compra sin algún tipo de recomendación directa es su meta y logran cumplir tal meta porque las sugerencias de compra de estos elementos son adaptables a los deseos y necesidades de los usuarios.
- Entender mejor lo que el usuario quiere es sin duda lo que muchos recomendadores de varios tipos de aplicación van consiguiendo, pues estos SR describen muy bien las preferencias de los usuarios que pueden retroalimentarse para generar futuras sugerencias a más usuarios.

2.1.1 Tipos de sistemas de recomendación

Existen varias técnicas para dar una recomendación, dependiendo del contexto, del tipo de producto y el objetivo general que se quiera cumplir, a continuación explicamos brevemente algunas de éstas técnicas:

- *Sistema de Recomendación Colaborativo*: También conocido como (Filtrado Social) es una técnica o método que realiza predicciones automáticas conocido como (filtrado) sobre los intereses de un usuario mediante la recolección de los gustos o preferencias de información de muchos usuarios (colaborador). En otras palabras el filtrado colaborativo se basa, en que si dos personas A y B tienen la misma opinión sobre un tema, entonces A tendría mayor probabilidad de tener la misma opinión que B en un tema distinto, que otra persona tomada al azar_(Jannach, Zanker, Felfernig, & Friedrich, 2011).
- *Sistema de Recomendación Demográfico*: Estos recomendadores basan su idea en las características demográficas que una persona da, como (edad, sexo, nivel de educación, domicilio, etc.), la idea general es que dicha persona tenga gustos similares con otras personas de características demográficas similares_(Gallardo, 2012).
- *Sistema de Recomendación de Contenido*: En estos sistemas se tiene información sobre las características (features) de cada producto o elemento a sugerir y se intentan extraer relaciones entre estas y la valoraciones de preferencia de un usuario cualquiera. Este sistema de recomendación no predecirá la valoración que un usuario realice, sino que asignará una puntuación a los productos, con esto se consigue indicar la idoneidad del producto para el usuario.
- *Sistema de Recomendación basado en Conocimiento*: Estos sistemas utilizan una base de conocimiento que describe cómo los distintos productos satisfacen las necesidades de un usuario y en qué medida (conocimiento funcional del entorno). De esta manera, el sistema encuentra el producto (o productos) que se ajusta a las necesidades que el usuario ha especificado. Esta búsqueda de productos a partir de las necesidades se realiza mediante un proceso de inferencia de algún tipo. Esta arquitectura permite a estos sistemas obtener y almacenar la información de distintas formas.
- *Sistema de Recomendación Híbrido*: Para mejorar los resultados de recomendaciones que podemos tener con uno de los recomendadores anteriores realizamos una hibridación de distintas técnicas, con la cual se pretende que el sistema obtenga todas las ventajas de los sistemas que se unieron y ninguna de sus desventajas, resultando así un recomendador más poderoso_(Jannach, Zanker, Felfernig, & Friedrich, 2011).

Según esta clasificación, a nuestro proyecto lo definiríamos como un Sistema de Recomendación de Hoteles en estado híbrido, pues en él podemos ver características de los SR basado en conocimiento y el de contenido, sin embargo estos matices no son suficientes para encasillarlo bajo esta definición, por ello no lo vinculamos a la clasificación vista

anteriormente, convirtiéndose en un recomendador más personalizado a las necesidades y objetivos que queremos alcanzar con este proyecto.

2.2 Investigación web

El campo de investigación fue principalmente la web ya que hemos empezado nuestro proyecto desde cero y como ayuda teórica hemos seguido el libro *Recommender Systems An Introduction* (Jannach, Zanker, Felfernig, & Friedrich, 2011), y *artículos publicados*. Esto nos ha llevado por un gran camino de búsquedas de aplicaciones parecidas en el mercado, su análisis y ahora comparación con lo que finalmente hemos creado lo miraremos en los siguientes apartados.

2.2.1 Sistema de Recomendación TripAdvisor-Trivago

TripAdvisor y Trivago son aplicaciones bastante completas las cuales bajo el modelo de gestor de contenido de los usuarios¹ aplican la recomendación y comparación de hoteles, hoy en día altamente usados en todo el mundo, su buscador de hoteles usa varios filtros importantes para personalizar la búsqueda del usuario de acuerdo a varios requerimientos que se consideran importantes desde su perspectiva, lo que quiere decir que estas aplicaciones han trabajado mucho ya que al igual que nuestro proyecto buscan la satisfacción de los usuarios, manejan una valoración de hoteles de acuerdo al status que la gente ha dado en cada comentario realizado, siendo el menor valor dos estrellas y el máximo cinco, en el caso de Tripadvisor y valoraciones de desde una estrella hasta cinco en Trivago, algo que también incluimos en nuestro sistema.

El refinamiento en cuanto a la cercanía, distancia o ubicación de hoteles en estas aplicaciones es configurable, sin embargo nosotros hemos diseñado e implementado una aplicación más sencilla y práctica tomando en cuenta que la distancia máxima sea de 0.5 km pensando en la preferencia de la gran mayoría de turistas con deseos de conocer los sectores del centro urbano de cada lugar que visiten, por comodidad, costes de transporte, desplazamiento y tiempo, esto explica que si un usuario selecciona un punto en el mapa periférico, fuera de urbe y simplemente un sitio sin edificaciones y vida social, no podrá encontrar hoteles cercanos en nuestra aplicación.

¹ Permite crear una estructura de soporte (*framework*) para la creación y administración de contenidos

2.2.2 Keywords y expresiones regulares

Al entrar en el análisis del contexto y el campo para el que trabajamos, parte de nuestra investigación trata de saber ¿qué se dice de los hoteles en la web?, buscando en lugares como foros (López, 2010), blogs, aplicaciones, revistas y catálogos que encierran al mundo hotelero en Madrid, nos encontramos con titulares de prensa, publicidad, opiniones, comentarios y hasta críticas de los usuarios que habían pasado por una experiencia buena o mala en alojamientos dentro del estado Español.

Esta investigación la complementamos con el siguiente análisis que detallamos a continuación.

Análisis de las palabras más usadas (keywords) por los usuarios para buscar un hotel

Hemos realizado un análisis de las palabras más usadas por distintos turistas, para encontrar un hotel en las ciudades más turísticas de España, para ello se ha hecho una clasificación por categorías y palabras claves que son las que más se escriben en los buscadores a la hora de solicitar un hotel por parte de los usuarios, toda esta información la recabamos de sitios de turismo, blogs de usuarios de la red y aplicaciones como: Tripadvisor², Trivago³ (lo más buscado), Booking⁴, NH Hotel Group⁵, entre otras webs de hoteles. Además hemos usado la herramienta Google Trends⁶ que sirve para conocer estadísticas de uso de las palabras claves permitiendo comparar volúmenes de búsqueda para distintas palabras y en diferentes ámbitos geográficos.

Los resultados obtenidos de la indagación para encontrar estas palabras claves con objetivo tal que nos sean útiles a la hora de manejar expresiones regulares que en este caso procesamos en nuestra aplicación son los siguientes:

Nombre de la categoría	término positivo	término negativo
general	ofertas, promociones, agradable, perfecto, genial, espectacular, encanto, inmejorable, recomendado, buenísimo, magnífico, excelente servicio, buena calidad, primer nivel, excelente hospitalidad, calidez, cordialidad, buen gusto, tranquilo, lujo, confort, mascotas	política de niños, política de mascotas, no niños, contaminación, malo en calidad precio, no cumple expectativas,

² <http://www.tripadvisor.es/Hotels>

³ <http://www.trivago.es/>

⁴ <http://www.booking.com/>

⁵ <http://www.nh-hoteles.es/hoteles/madrid>

⁶ <https://www.google.es/trends/>

		instalaciones defectuosas, mal funcionamiento, mala calidad
ubicación	céntrico/centro/cerca de, transportes públicos/ bien comunicado,	aeropuerto, periférico, mala ubicación,
precio/costo	barato/accesible/gama media	excesivo, caro
limpieza	limpio, limpieza	sucio, suciedad
servicios	wifi, tv cable, aire acondicionado, calefacción, enchufes, parking, acceso a sillas de ruedas (accesibilidad), piscina, gimnasio, spa, reuniones, celebraciones	pésimo desayuno, mala atención, deplorables, tercermundista, no es de N estrellas, pésimo spa
habitaciones	cómodas, insonorizadas, buena cama, cama extra, cama doble, caja fuerte	incómodas, obligación cama supletoria, sin cama supletoria, baño pequeño, baño mala condiciones, mal mantenimiento, remodelación
bar/restaurant	desayuno gratuito, buena comida/cocina, rapidez, comida de calidad	mal servicio de desayuno, pésimo café, no hay bebida de bienvenida, desayuno caro

Tabla 2.1 Palabras claves

2.2.2.1 Keywords y expresiones regulares positivas

Las keywords y expresiones regulares positivas las definimos como una categorización que servirá a dar un cálculo a favor de la popularidad de un cierto hotel, estas serán las felicitaciones y agradecimientos de los usuarios, como ejemplo se lista a continuación palabras que usamos en nuestro diccionario para escanear y comparar cada web analizada.

Lista de expresiones regulares positivas: en España

- con spa y gimnasio
- desayuno gratis
- wifi en zonas comunes
- wifi gratuito
- con wifi en todo el establecimiento
- limpieza continua
- con piscina cubierta
- con zona de cocina
- con terraza
- con lavandería
- con restaurante buena comida
- parking en el hotel
- bien comunicado
- cerca de estaciones de transporte
- en el centro
- céntrico
- bien situado
- habitación con abundantes enchufes
- habitaciones cómodas
- habitación insonorizada
- tarifa plana mini bar
- barato sencillo
- magnífico precio
- buena relación calidad precio
- con ofertas y promociones
- con mascotas permitidas
- estancia agradable
- con excelente hospitalidad
- atención rápida
- buena calidad
- elegante y distinguido
- buen diseño y estilo
- buena experiencia
- con encanto

- recomendado experiencia inmejorable
- perfecta estancia
- sentirse como en casa
- muy buena opción
- trato muy cordial
- buenísimo servicio
- todo lo que necesitas

Ejemplo:



Figura 2.1 Comentario de un usuario usando palabras positivas

2.2.2.2 Keywords y expresiones regulares negativas

Las keywords y expresiones regulares negativas las definimos como una categorización que servirá a dar un cálculo en contra de la popularidad de un cierto hotel, estas serán las críticas (Alfredo, 2013), las denuncias y opiniones negativas de los usuarios como ejemplo listamos a continuación palabras que usamos en nuestro diccionario para escanear las webs e iniciar los cálculos.

Lista de expresiones regulares negativas: en España

- obligan a contratar cama supletoria para niños menores de 4 años
- política de niños
- prohíben hospedaje de mascotas
- sin posibilidad de cama supletoria
- contaminan el medio ambiente

- baño muy pequeño
- baño con poca presión de agua
- hotel mal ubicado sector muy ruidoso
- no tiene parking
- mala relación calidad precio
- muy mala atención
- checkin lento
- sin organización de eventos
- mal trato por personal del hotel
- mala opción
- por debajo de mis expectativas
- es un chasco
- mala recepción
- baño poco cuidado
- instalaciones en mal funcionamiento
- habitaciones sin mantenimiento
- necesita mejorar
- remodelación de habitaciones
- relación calidad precio deplorable
- servicios de la habitación son tercermundista
- no cumple el número de estrellas
- calidad y servicio no es de N estrellas
- spa y ubicación pésimos
- spa pequeño
- desayuno muy caro
- jacuzzi averiado

Ejemplo:



Figura 2.2 Comentario de un usuario usando palabras negativas

2.3 Información del contexto y del usuario

Muchos de los sistemas de recomendación que podemos encontrar en la actualidad tienen por debajo la categorización de los usuarios, para ello crean perfiles de acuerdo a algunos parámetros como pueden ser las distintas demandas de servicios por cada individuo, apoyando a esta estructura los gustos y preferencias de los usuarios, que a la vez se suman a las tendencias del mercado hotelero y el avance tecnológico. Dichos perfiles se van componiendo de filtros, también llamados “features” que dan la pauta para especificar un perfil, con estas categorías se trabaja internamente para dar un peso a las que más o menos se ajusten al perfil deseado. Todo este proceso de clasificación de información sirve de mucho para refinar la recomendación y dar un resultado bastante ajustado al que realmente demanda el usuario.

Los usuarios podrán utilizar nuestro SRH en diversos contextos tales como:

- *Cientes que buscan un hotel:* los clientes de nuestra aplicación podrían entrar en cualquiera de los siguientes campos:
 - Estático: para usuarios que deciden hospedarse en algún hotel de España, su tarea sería seleccionar la ciudad a visitar y luego en ella buscar los hoteles cercanos al centro de dicha ciudad para luego con la flexibilidad que disponen

ir evaluando los hoteles de interés bajo los analizadores estadístico y sentimental que proporcionados.

- Dinámico: dirigido a usuarios que gracias a la interfaz con el mapa de nuestra web pueden seleccionar un punto en el mapa, el que deseen y buscar hoteles cercanos al rededor de aquel punto y seguir con la valoración de los hoteles como en la parte anterior.
- Movilidad: para un servicio como el que damos no podíamos dejar de lado un geo localizador en el mapa que si el usuario lo activa indicará su posición actual y bajo esta se podrá buscar hoteles cercanos a evaluar y futuramente recomendados y sea el usuario quien tome la decisión de elegir el más apto a sus necesidades.

Los usuarios que se definan dentro de cualquier campo explicado antes, entran directamente a realizar la evaluación de los hoteles que consideren oportunos, para ello tienen a su servicio los dos analizadores que tanto hacemos mención en este proyecto.

3 Explicación Teórica

Para realizar el análisis de la información hemos desarrollado nuestro modelo de datos sobre el cual vamos a estudiar, evaluar y analizar para así llegar a una valoración, la cual determinará qué tan bueno es un hotel. De esta manera nuestro modelo se centra en los comentarios de un hotel que se realizan en diversas plataformas sociales, encontrados por el buscador Google, para así una vez obtenida la información y tratada en tiempo real, puedan enviarse los resultados a la aplicación web.

3.1 Sistema de Recomendación de Hoteles

El turismo y el sector hotelero son de los sectores más importantes y dinámicos en el negocio de consumidores, hacen que el comercio electrónico crezca como lo hacen diariamente las transacciones online de forma sorprendentemente rápida. La mayor tasa de negocios a nivel mundial es el mercado turístico, países desarrollados se han dado cuenta de esto hace mucho y trabajan para mejorar cada vez los recursos y resultados en la web que hoy se la define como fuente universal de información y servicios para las personas. Esta área da para mucha investigación porque no ha sido totalmente explotada, la información relevante a este campo es abundante y sufre alteraciones en el tiempo, lo que la hace digna de estudio. Los sistemas de recomendación están aprovechando la convergencia de la Web Semántica, la Inteligencia Artificial, la computación Ubicua (Con espacios Inteligentes sensibles al contexto), las redes inalámbricas y móviles con el objetivo de personalizar y mostrar recomendaciones de forma más precisa y menos perceptible al usuario_(Fuentes, 2010).

El usuario actual requiere de muchos y eficaces servicios, tiene peticiones complejas y específicas, no quiere perder el tiempo y le importa mucho satisfacer todas sus necesidades.

Nuestro proyecto quiere ambiciosamente cubrir estas necesidades, dar una ayuda más específica y eficiente al usuario, de modo que los resultados disipen sus frustraciones y eliminen dudas en cuanto a un real filtro de búsqueda cuyas salidas serán las mejores recomendaciones para las preferencias de alojamiento en España de cada individuo.

El sistema de recomendación hotelero que hemos desarrollado tiene una interfaz de usuario amigable y deducible donde el usuario se sentirá cómodo utilizándolo. Nuestro proyecto tiene bien definido tres servicios los cuales se acoplan y tienen relación directa con el resultado que será dar una recomendación y valoración al usuario de los hoteles que sean de su interés en España, para hacer más fácil el detalle de estos servicios los enumeramos a continuación:

1. Búsqueda de Hoteles Cercanos: Una aplicación como ésta no puede dejar pasar la geo localización como parte e información dinámica que se puede controlar para beneficio

de la aplicación y el usuario, varias formas de que el usuario busque hoteles cercanos a una posición actual se verán a continuación:

- Geo localización: Una vez el usuario permita a la herramienta usar su localización, a partir de ésta el sistema le recomendará un número específico de hoteles cercanos, gracias a la tecnología Google Maps que es cómplice de dicho servicio.
- Definir localización: Otra forma básica y muy simple de definir una ubicación será que el usuario defina un punto en el mapa y la tomaremos como localización de la cual podemos recomendar hoteles.
- Búsqueda por ciudad: Damos al usuario la opción de buscar directamente la ciudad que le despierte curiosidad turística para conocer dentro de España al seleccionarla, después de esto se puede recomendar directamente los hoteles cercanos o el usuario puede hacer uso de cualquiera de los dos casos anteriores adicionalmente. Especificamos en este punto que tenemos 16 ciudades españolas, esto no debe limitar en nada al sistema, pues si el usuario se encuentra en otra ciudad de España o el usuario quiere alojarse en una distinta de las que se encuentran en las opciones lo puede hacer navegando directamente en el mapa que ofrece la aplicación o poner directamente el nombre del hotel que quiere evaluar y el sistema funcionará de igual forma, el porqué de poner solo 16 ciudades lo explicamos con un análisis e investigación que hicimos tal que así. Limitamos el número de ciudades por seleccionar a las ciudades con mayor demanda turística en España en las distintas categorías de turismo como son el religioso, gastronómico, cultural, de entretenimiento, etc. y en el ranking de las ciudades más visitadas figuran las siguientes:

- Alicante
- Barcelona
- Bilbao
- Burgos
- Donostia - San Sebastián
- Girona
- Granada
- Madrid
- Palma de Mallorca
- Salamanca
- Santa Cruz de Tenerife
- Santiago de Compostela
- Sevilla
- Toledo
- Valencia
- Zaragoza

2. **Analizador Estadístico:** Nuestro sistema de recomendación consta de una parte meramente estadística, en la que manejamos resultados medios de la búsqueda de una combinación de queries, estas queries hacen referencia a distintas formas de búsqueda para un mismo elemento, que en concreto están relacionadas con la búsqueda de un hotel. Usando para ello propiedades estadísticas de probabilidad condicionada como el Teorema de Bayes y manejando relaciones entre términos positivos y negativos de las queries, esto nos genera un resultado cuyo valor será normalizado para expresarlo en una valoración sobre 5 para un ranking de estrellas y favorecer el feedback al usuario, representando la valoración estadística del hotel analizado, esta valoración se convierte en el campo de comercio como la reputación del hotel estudiado.
3. **Analizador Sentimental:** Uno de los recomendadores más complejos que hemos desarrollado por el manejo del lenguaje natural, para la realización de este recomendador tuvimos que especificar keywords y expresiones regulares que en concepto son las palabras más usadas en las consultas de hoteles en lenguaje castellano añadiendo a los mismos términos en inglés como por ejemplo "wifi", "spa", que define algunos servicios que el usuario requiere. Estas palabras fueron analizadas bajo el escaneo de comentarios publicados por los usuarios en webs tales como blogs, foros, comentarios en redes sociales, también hemos hecho uso la herramienta de Google Analysis Tender para validar las keywords y expresiones regulares elegidas. La idea general es hacer un análisis emocional con el hotel elegido, relacionando positiva y negativamente los resultados que vayamos obteniendo para calcular una valoración normalizada a cinco estrellas como lo hicimos anteriormente, para dar al usuario la reputación sentimental del hotel seleccionado.

3.1.1 Analizadores

Analizador Estadístico:

Da una recomendación de la forma más flexible posible para el usuario, el cual es libre de configurar su búsqueda con los tres filtros principales que son: el precio, los servicios y la ubicación.

Precio: Será un factor importante ya que si se ha seleccionado para la valoración indicará en porcentaje cuan bueno o malo es, siendo a menos porcentaje muy caro y a más porcentaje el precio es cada vez más cómodo o barato, la relación calidad precio se verá en la valoración total ya que esta depende directamente de los tres filtros a evaluar.

Servicios: Otro de los filtros son los servicios, valorados en general y evaluados como buenos servicios con un porcentaje alto en la valoración resumida o individual y malos servicios con porcentajes pobres.

Ubicación: Tenemos la ubicación como final de los filtros, importante sobre todo para cuando el usuario se encuentra dentro del campo de movilidad y será desde luego mayor su porcentaje si la cercanía del hotel es alta y todo lo contrario si es un hotel periférico o demasiado alejado.



Figura 3.1 Captura de pantalla del analizador estadístico

Además ofrecemos una valoración basada en un analizador sentimental que mostrará información complementaria.

Analizador Sentimental:

Recomendación sentimental social dado que este analizador irá en búsqueda de los comentarios hechos por los clientes de los hoteles para fijar de acuerdo a nuestro diccionario de datos la puntuación que merece el hotel en juego, este analizador nos dará una valoración sentimental total, pero además el usuario podrá contrastar la valoración del analizador anterior gracias a que mostramos los resultados parciales de las webs en detalle clasificándose como listas de palabras positivas y negativas para una reconstrucción de las opiniones sociales del hotel seleccionado en la web, ¿qué se dice o se habla del hotel? de esta manera el usuario tiene información complementaria para ver la popularidad del hotel y tomar sus propias decisiones.



Figura 3.2 Captura de pantalla del analizador sentimental

web: <http://www.petitpalace.com/hotel-santa-cruz-en-sevilla/>

RESULTADOS

Lista de palabras positivas (+)	
wifi gratis en el	2
wifi gratis, bual	1
tarifa no reembolsa	1
Oferta última hora	1
Cama de matrimonio	1
ofertas ni promocione	1
wifi gratis... HD	1
cama de matrimonio	2
Oferta especial des	1
wifi gratis, TV,	1
Precio desde: 63.0 €	2
desayuno buffet de 07	2
Precio desde: 99.0 €	1
Oferta 10% de descu	1
Precio desde: 70.0 €	1
tarifa normal de un	1
Precio desde: 59.5 €	2
wifi gratis, TV,	1
Precio desde: 62.0 €	1
Oferta No reembolsa	1
precio garantizado	1
wifi gratis, baño	1
wifi gratis... Esta	1
precio en el Hotel	1
desayuno buffet gratu	1
wifi gratis, gran	1
wifi gratis, duch	1
magnífica Habitación Ind	1
Oferta Última Hora	1
Lista de palabras negativas (-)	
sin intermediarios y	1
mal de una habitació	1
Lista de expresiones regulares positivas (+)	
wifi gratis	10

Figura 3.3 Captura de pantalla de los resultados del analizador sentimental

- *Desarrolladores que deseen hacer uso de nuestro SRH*

Nuestro SRH puede dar servicios a otras aplicaciones multiplataforma, podemos conectar y adaptarnos a otros servicios y ser parte de un sistema más complejo, es en este punto de nuestro trabajo donde hablaremos de un ejemplo real.

Hemos realizado un módulo de servicio para una aplicación de ocio como es Madrid Manía, buscamos la forma de acoplarnos a sus necesidades, dadas las demandas y requerimientos de los integrantes del grupo realizamos la conexión para adaptar nuestro SRH a un servicio que prestará Madrid Manía, desarrollamos dicha adaptación que explicamos con mayor detalle de la siguiente manera:

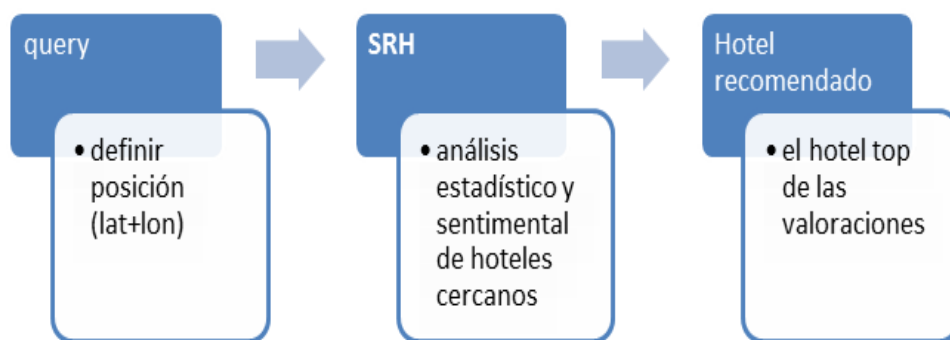


Figura 3.4 Esquema del servicio ofertado a Madrid Manía

Query: Define la posición en la cual se requiere un servicio de recomendación, se dará un punto geolocalizado por una latitud y longitud, para obtener el centroide a partir del cual se recomendará el mejor de los hoteles cercanos.

Módulo SRH: Esta parte de nuestro proyecto da el servicio de Recomendación Hotelera a Madrid Manía para que lo pueda usar en uno de sus servicios.

Nuestro módulo tendrá como entrada la query anteriormente explicada, con la cual hará uso de recomendadores internos implementados en nuestro proyecto como son el *estadístico* y *sentimental*, haremos un análisis de las valoraciones devolviendo el mejor hotel valorado, esto será devuelto como salida vía JSON que llevará la información del hotel recomendado, su formato lo explicamos a continuación.

Hotel Recomendado: Representa la salida de nuestro módulo y es un JSON con información del nombre del hotel y su puntuación o valoración dentro del rango de [1-5] normalizada así por manejar un ranking de estrellas muy popular para hoteles

Hemos realizado un **módulo** para que Madrid Manía pueda realizar pruebas, sin que sus desarrolladores pierdan su tiempo en esperar la respuesta de nuestro sistema, y así facilitarle su trabajo.

De esta manera concluimos que nuestro proyecto puede ser escalable y cumplir con objetivos como la adaptación a sistemas para complementar servicios o simplemente evolucionar nuestro propio SR.

3.2 Características Recomendador de Hoteles

Nuestro sistema ofrece un servicio que lo diferencia de muchas otras aplicaciones, tales como TripAdvisor y Trivago, que solo se centran en recopilar información de sus clientes, y no en analizar la masiva cantidad de datos disponible en Internet.

Al día de hoy no existe un servicio que sea capaz de ofrecer un determinado hotel en base a los comentarios, críticas o elogios de los clientes que publican en diversos foros y blogs ajenos a sus propias bases de datos. Es aquí donde nosotros con nuestro sistema intentamos dar ese servicio.

3.3 Perfil del usuario

Nuestra aplicación se adapta tanto a los usuarios que valoran un hotel, bien sea por su precio, por la cantidad, calidad de servicios que dispone y por la zona en la que se encuentra ubicada.

Destacamos tres perfiles de usuarios.

1. Usuario dependiente del coste
2. Usuario en busca de la comodidad
3. Usuario inquieto

Usuario dependiente del coste: Es el cliente que busca un hotel en relación a lo que puede pagar por hospedarse en el mismo.

Usuario en busca de la comodidad: Es el cliente que busca un hotel con la máxima o mínima cantidad de servicios que quiere disponer.

Usuario inquieto: Son todos los clientes que buscan un alojamiento que esté ubicado lo más cerca posible de las zonas más visitadas de la ciudad a la que quieren conocer.

3.4 Categorización de keywords y expresiones regulares

Teniendo en cuenta las diferentes categorías de usuarios, hemos adaptado nuestra aplicación en base a ello y es por eso que para el análisis e interpretación de los comentarios publicados en las diferentes webs hemos optado en separarlos en diferentes categorías, las mismas son:

1. ubicación
2. precio
3. servicios

Por comodidad solo se detallarán algunas subcategorías de cada categoría, con sólo algunos términos positivos y negativos para hacer más fácil su interpretación.

Servicios

	término positivo	término negativo
Gym	amplio	malo, pequeño, poco agradable ...
Spa	accesible	no dispone, horario no continuo
Wi-fi	máxima velocidad, gratis	lento, con coste

Tabla 3.1 Ejemplos de subcategorías de Servicios y sus términos

Precio

	término positivo	término negativo
Gym	amplio	malo, pequeño, poco agradable ...
Spa	accesible	no dispone, horario no continuo
Wi-fi	máxima velocidad, gratis	lento, con coste

Tabla 3.2 Ejemplos de subcategorías de Precio y sus términos

Ubicación

	término positivo	término negativo
Distancia	bien comunicado, cercana al centro, próxima a los museos,	alejada
Próximo	accesible	no dispone, horario no continuo

Tabla 3.3 Ejemplos de subcategorías de Ubicación y sus términos

Como se puede observar, en las tablas, cada categoría está compuesta por subcategorías en las que se detallan los términos positivos y negativos que fueron obtenidos del amplio análisis y estudio de diversas webs (blogs, foros y recomendadores de hoteles).

Dichas tablas fueron imprescindibles para el desarrollo de nuestro *analizador sentimental*, ya que nos permite valorar las características de los hoteles para con ello más adelante realizar una estimación de cuán bueno es realmente, o si está muy infravalorado o sobrevalorado.

3.5 Sistema de valoración

Los resultados que representarán el valor que ha obtenido un determinado hotel tras haberlo analizado, tanto sentimental como estadístico, se puntuará de 0 a 5 ambos incluidos, ya que dicha puntuación es estándar en los recomendadores existentes.

También mostraremos los resultados parciales, por categorías, evaluados con un valor porcentual.



Figura 3.5 Captura de pantalla de nuestro sistema de valoración

3.6 Algoritmos

Una vez que nos hemos centrados solo en los comentarios disponibles en las webs, el siguiente paso fue decidir cómo tratar dicha la información, la cual nos ayudará a analizar tanto las webs encontradas como las expresiones obtenidas y de esa manera poder obtener unos resultados más precisos a la hora de puntuar los hoteles, para ello aplicamos dos estrategias:

- Búsqueda estadística.
- Búsqueda sentimental.

3.6.1 Estadístico vs Sentimental

Con este algoritmo obtenemos un resultado estadístico cuya obtención proviene de un cálculo estadístico que utiliza el número de matching de nuestra búsqueda con las webs encontradas por Google⁷.

⁷ <http://es.wikipedia.org/wiki/Google>



Figura 3.6 Captura de pantalla de una búsqueda avanzada en Google

Con este cálculo pretendemos obtener la probabilidad de que un determinado hotel tenga diversas características relacionadas con las categorías Precio, Ubicación y Servicio

Para ello hemos distribuido las diferentes categorías por términos:

- Ubicación
 - buena
 - mala
- Servicios
 - buenos
 - malos
- Precios
 - barato
 - caro

Para calcular la valoración tanto a nivel porcentual como de estrellas hemos aplicado el siguiente algoritmo el cual incluye cálculos probabilísticos.

Nota: Debido a que el buscador de Google nos devuelve un determinado resultado en función de la query enviada, y después de realizar varios casos de pruebas, hemos decidido realizar 3 solicitudes con queries distintas por cada término.

Por ejemplo: para la categoría ubicación con término bueno se envían las siguientes:

1. q1: “buena ubicación”
2. q2: “buena ubicación *”
3. q3: “buena ubicación ”

y así sucesivamente para las restantes categorías y términos correspondientes, finalmente solo nos quedamos con el máximo de las tres, $\max(q1, q2, q3)$, este máximo será quien dé el valor final tanto para B, A y $A \cap B$ con los cuales trabajaremos para el desarrollo de dicho cálculo.

Pero debido a la limitación en cuanto al tiempo de procesamiento optamos por solo evaluar la primera query de cada categoría.

término+: término positivo.

término- : término negativo.

B: <término+><término-><Categoría><Ciudad><nombre del hotel>

A: <términoX><Categoría><Ciudad><nombre del hotel>

términoX: término+ o término-

$A \cap B$: <Categoría><términoX><Ciudad><nombre del hotel>

$$P(A|B) = P(A \cap B) / P(B)$$

$P(A \cap B)$ y $P(B)$ son números de resultados de búsquedas obtenidos por Google

$P(A|B)+$: Probabilidad sobre términos positivos

$P(A|B)-$: Probabilidad sobre términos negativos

R: Resultado **positivo**, obtenido una vez efectuados cálculos con $P(A|B)+$ y $P(A|B)-$,

$$P(A|B)+ > P(A|B)- \rightarrow R = (1 - (P(A|B)- / P(A|B)+)) * 100$$

$$P(A|B)- > P(A|B)+ \rightarrow R = (P(A|B)+ / P(A|B)-) * 100$$

A continuación damos a conocer un diagrama que ayudará a comprender mejor el cálculo de R, ya que se pueden dar diversos casos.

Para simplificar la sintaxis hemos representado los siguientes términos antes mencionados como:

$$RP = P(A|B)+$$

$$RN = P(A|B)-$$

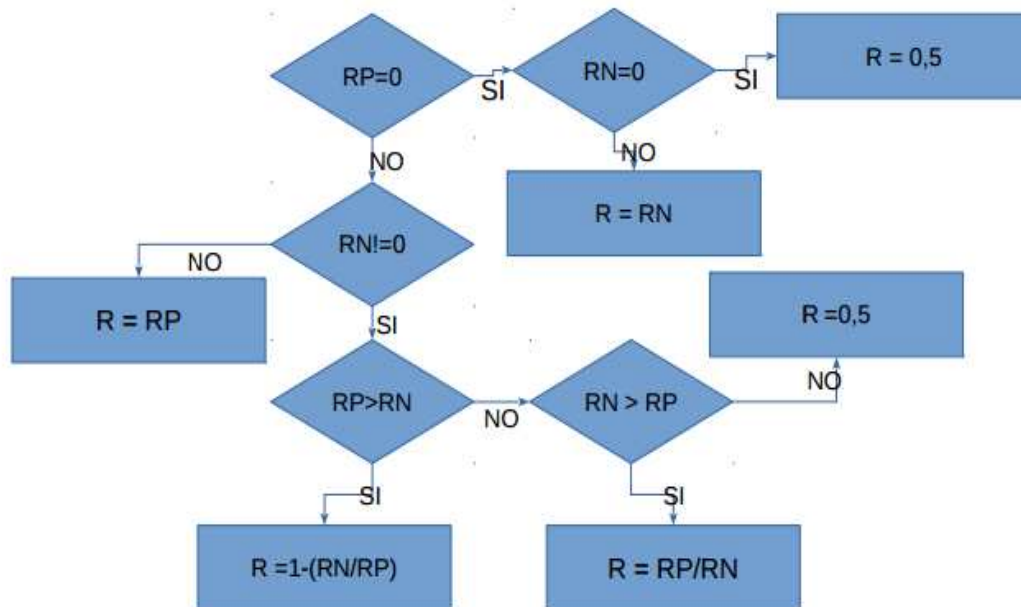


Figura 3.7 Diagrama de estados del algoritmo estadístico

Citamos un ejemplo en el que queremos calcular la probabilidad parcial y total del hotel Petit Palace tanto en servicio, ubicación y precio.

B: Ubicación * Hotel Madrid Petit Palace.

A: **buena** Ubicación Hotel Madrid Petit Palace.

$A \cap B$: buena Ubicación Hotel Madrid Petit Palace.

Resolvemos $P(A|B)+ = 125000 / 180000 = 0.6944444$.

B: Ubicación * Hotel Madrid Petit Palace.

A: **mala** Ubicación Hotel Madrid Petit Palace.

$A \cap B$: buena Ubicación Hotel Madrid Petit Palace.

Resolvemos $P(A|B)- = 38300 / 180000 = 0.2112$.

$P(A|B)+ > P(A|B)- \rightarrow R = (1 - (0.2112/0.6944444)) * 100 = 69.36 \%$ de que este bien ubicado.

Aplicamos el mismo procedimiento tanto para Precio y Servicio.

Finalmente calculamos la media tanto para la cantidad en porcentaje y en estrellas.

n : cantidad de categorías evaluadas

R_i : Hace referencia al resultado positivo de porcentaje de la categoría i -ésima .

S_i : Hace referencia al resultado positivo de estrellas de la categoría i -ésima.

$$\text{TotalResult} = \frac{1}{n} \sum_{i=1}^n R_i$$

$$\text{TotalResult} = \frac{1}{n} \sum_{i=1}^n S_i$$

(*): Damos por hecho que Google realiza una búsqueda de la <categoríaX> sin contemplar si sus términos son positivos o negativos.

Dado que R nos devuelve resultados que contemplan términos positivos, hemos decidido como de positivo es, para ello hemos establecido determinados márgenes que dependen de la cantidad de estrellas que ha obtenido el hotel deseado una vez evaluada la característica que el usuario haya establecido.

Stars < 2 → **malo**.

2 < Stars < 3.3 → **bueno**.

Stars > 3.3 → **excelente**.

Estos márgenes son establecidos tanto para precio, servicio o ubicación.

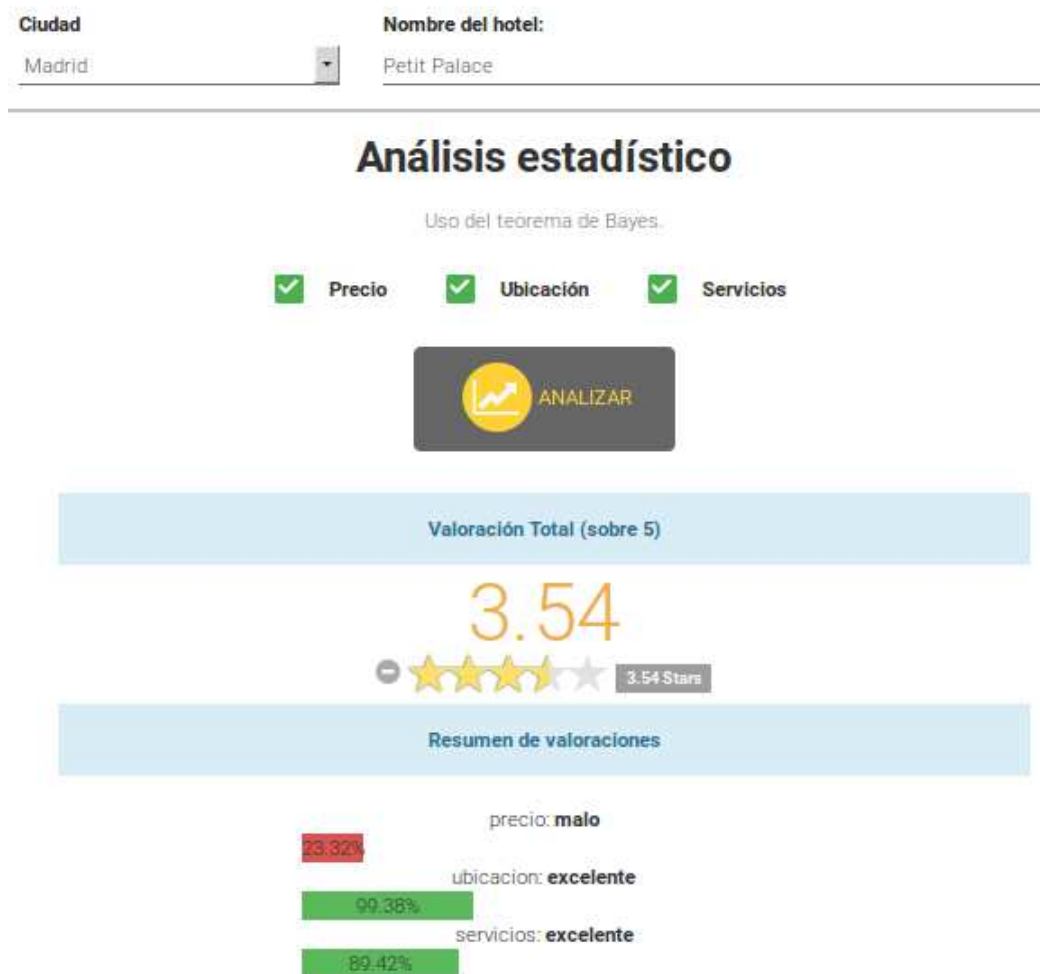


Figura 3.8 Captura de pantalla del resultado estadístico obtenido

3.6.2 Búsqueda Sentimental

Mediante la búsqueda de las expresiones y comentarios de los usuarios satisfechos o no con los servicios que les ha ofrecido el hotel en el que han permanecido alojados durante su visita a España hemos procedido a analizarlos de la siguiente manera:

La tablas reflejadas arriba expuestas nos sirven como diccionario para obtener los resultados para nuestro estudio, dichos resultados lo representamos de la siguiente manera:

expPS: Expresiones positivas.

nexpPS: Total de expresiones **positivas**, correspondientes a **servicios**, almacenadas en nuestro diccionario.

encontrado(expPS): Función que dada un expresión regular positiva retorna la cantidad de matching obtenidas.

NREPS: Cantidad que refleja el número de matching obtenido en las diversas webs analizadas según expPS

$$NREPS = \sum_{expPS}^{nexpPS} encontrado(expPS)$$

expNS: Expresiones negativas.

nexpNS: Total de expresiones **negativas**, correspondientes a **servicios**, almacenadas en nuestro diccionario.

encontrado(expNS): Función que dada un expresión regular retorna la cantidad de matching obtenidas.

NRENS: Cantidad que refleja el número de matching obtenido en las diversas webs analizadas segunexpNS

$$NRENS = \sum_{expNS}^{nexpNS} encontrado(expNS)$$

Aplicamos la misma metodología para el cálculo de las siguientes

- *NREPP* : Cantidad de expresiones regulares **positivas** según **precio**.
- *NRENP*: Cantidad de expresiones regulares **negativas** según **precio**.
- *NREPU*: Cantidad de expresiones regulares **positivas** según **ubicación**.
- *NRENU*: Cantidad de expresiones regulares **negativas** según **ubicación**.
- *PW*: Cantidad de palabras positivas.
- *NW*: Cantidad de palabras negativas.

Cálculo final:

PRE: Cantidad total de las expresiones regulares positivas.

NRE: Cantidad total de las expresiones regulares negativas.

$$PRE = NREPP + NREPS + NREPU$$

$$NRE = NRENP + NRENS + NRENU$$

$$TP = (PRE + PW) * pp$$

$$TN = (NRE + NW) * pn$$

pp y pn son los pesos que se aplican tanto a las partes positivas como negativas.

RPN : Resultado final obtenido una vez aplicado el algoritmo reflejado en el siguiente diagrama.

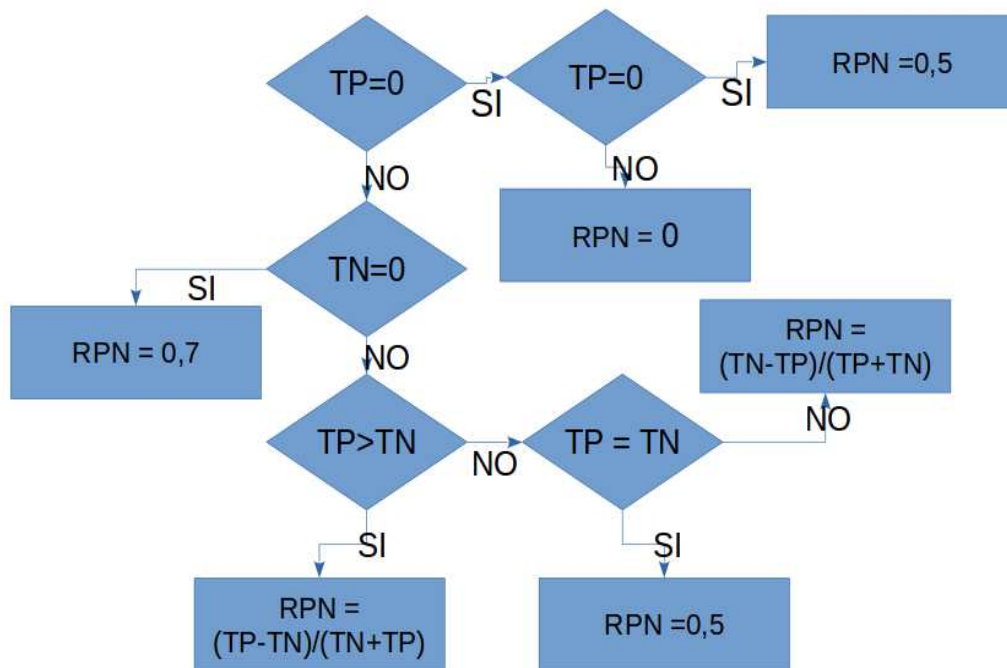


Figura 3.9 Diagrama de estados del algoritmo sentimental

3.7 Conclusiones

Como conclusión afirmamos que el estudio de los datos obtenidos fueron un gran reto para nosotros ya que la obtención de los mismos no fue nada fácil, obligándonos a realizar una investigación sobre las conexiones Proxy, que explicamos en el capítulo 6 y como poder realizar las mismas mediante la red **TOR** y así poder realizar peticiones a Google sin que él mismo nos limite.

El análisis de los datos fue un gran reto para nosotros, obligándonos a utilizar ciertos conceptos estadísticos los cuales nos permitieran obtener resultados fiables, resultados a los cuales comparamos con los dados por TripAdvisor y Trivago.

Justamente lo anterior nos obligó a refinar los algoritmos e ir retroalimentando nuestro sistema con pruebas, comparaciones y análisis, con lo cual podemos decir que el ensamblaje fue un poco costoso en cuanto al tiempo que dedicamos.

4 Implementación

Finalizado el estudio teórico de la información extraíble de diversas webs antes mencionadas, se ha diseñado e implementado un backend con la funcionalidad necesaria para responder a los objetivos planteados en el capítulo 1.2 relativos a la extracción, análisis y visualización de información. Este backend ha de ser capaz de extraer, analizar la información mediante dos estrategias de tratamiento y procesamiento de la información, y por último aportar las facilidades necesarias para que toda esta información tratada pueda ser visualizada a través de nuestra aplicación y por aplicaciones realizadas por terceros como Madrid Manía.

4.1 Arquitectura Backend

En nuestro backend determinamos cuáles serían las estrategias de manipulación de la información, explorando diversos foros, blogs, etc. Establecimos para ello nuestro modelo de datos que consta técnicamente en las expresiones tanto positivas como negativas encontradas, así como el número de resultados encontrados por Google (matching de palabras) a los que aplicamos técnicas estadísticas.

También almacenamos un diccionario de palabras y expresiones regulares tanto positivas como negativas.

4.1.1 Funcionalidad y diseño

Nuestro Backen está capacitado para realizar las siguientes funcionalidades:

Conectarse a Google mediante su API *Google Places API Web Service*⁸: Esta conexión nos permite extraer los 20 hoteles más cercanos de una determinada posición y radio, manipulamos dicha información para solo retener los datos que nos interesa como: latitud, longitud, nombre del hotel y dirección.

Conectarse a Google a través de la Api *Google Search*⁹: Con esta conexión obtenemos las el contenido de las webs que analizaremos en busca de todo tipo de comentarios sobre un hotel determinado.

Conectarse a Google¹⁰ sin ninguna Api: Debido a las limitación que nos brinda Google en cuanto al número de peticiones a través de su api pública, nos hemos visto obligado a obtener

⁸ <https://developers.google.com/maps/documentation/geocoding/>

⁹ <https://developers.google.com/web-search/docs/>

información mediante una conexión directa en la cual la información obtenida se encuentra en formato html, lo realizamos para obtener el número de match que Google encuentra en las webs sin tener ninguna limitación en cuanto al número de conexiones con su api.

Envío de información al cliente: Una vez analizada y gestionada la información, en el backend, procedemos a realizar el envío de resultados en formato JSON¹¹, siendo la forma más tradicional empleada por la mayoría de los servicios WEB concretamente los que hacen uso de la arquitectura REST.

La implementación del Backend fue realizada con Java, junto con el apoyo diversas librerías tales como:

Jersey¹²: Nos ofrece la posibilidad de realizar una comunicación REST entre el Backend y el Frontend.

Gson¹³ y JSON: Para estructurar la salida en formato JSON estándar por Google

Jsoup¹⁴: Utilizada para eliminar las etiquetas del HTML¹⁵ obtenido y analizar solo su contenido de forma más rápida.

Poi¹⁶: Librería que nos facilita el paso de los datos a hojas Excel para desde allí poder realizar el análisis y test de nuestra aplicación.

El siguiente esquema representa la arquitectura del backend

¹⁰ <https://www.google.com/>

¹¹ <http://json.org/>

¹² <https://jersey.java.net/>

¹³ <https://code.google.com/p/google-gson/>

¹⁴ <http://jsoup.org/>

¹⁵ <http://www.w3schools.com/html/>

¹⁶ <https://poi.apache.org/>

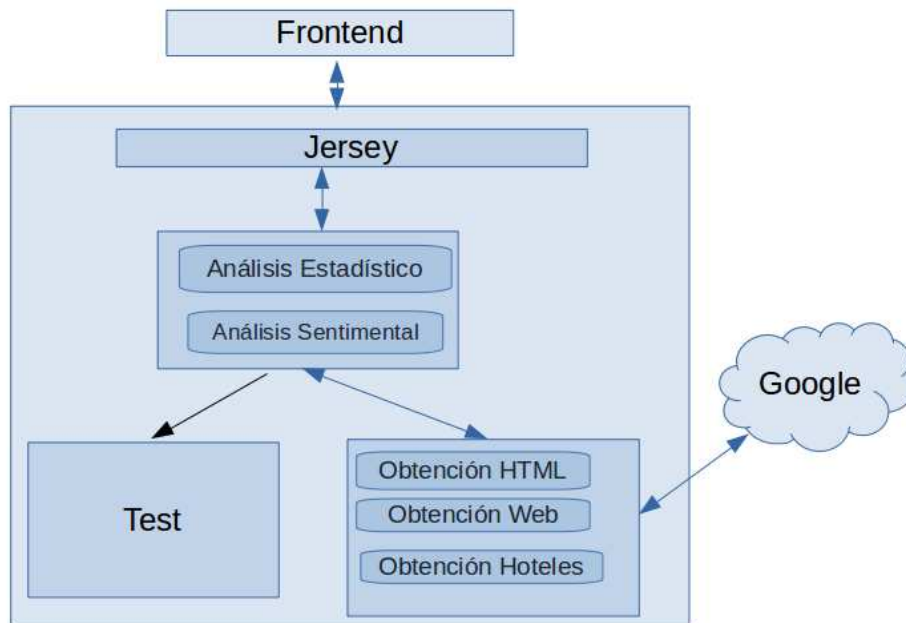


Figura 4.1 Arquitectura backend

4.1.2 Tecnologías

A lo largo de nuestro proyecto fuimos investigando qué tipo de tecnologías podríamos utilizar y cuáles eran las que se adaptarían mejor a nuestras necesidades, tecnologías como JSP¹⁷ (Java Server Page) y Node.js que eran candidatas a utilizarlas. Luego de adentrarnos un en la investigación de las mismas y viendo la dificultad que tendríamos a la hora de poder adquirir los conocimientos necesarios para poder desarrollar nuestro sistema y el tiempo que le dedicaríamos, fueron motivos suficiente para descartarlas y buscar otras alternativas.

4.1.3 Tecnologías utilizadas

Finalmente debido al tiempo y a los conocimientos que tenemos hemos decidido optar por Java la cual acompañada con Jersey nos permitía la posibilidad de ofrecer un servicio REST, enviando los resultados en formato JSON¹⁸ y con a la api Gson poder darle un estilo.

¹⁷ <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>

¹⁸ <http://json.org/json-es.html>

4.1.3.1 Java

Java, uno de los lenguajes más populares de la actualidad que tiene la siguiente filosofía "Write once, runAnywhere" es uno de los lenguajes que ha influido mucho en el ámbito de la programación.

Según la compañía TIOBE que se especializa en la evaluación y el seguimiento de la calidad del software, ponen a Java como una de las tecnologías más utilizadas_(TIOBE, 2015).

Teniendo en cuenta dicho análisis y la experiencia que tenemos con el lenguaje fueron los motivos por los que hemos elegido Java como base de nuestro Backend.

4.1.3.2 Jersey

Con la existencia de varias implementaciones para realizar la comunicación REST¹⁹ entre cliente y servidor:

1. Jersey
2. RestEasy²⁰
3. Restlet²¹
4. Apache CXF²²

Hemos optado finalmente por Jersey, alguno de los motivos son los siguientes:

- Muy utilizado y **sencillo**
- Soporta comunicaciones asíncronas (web sockets²³)
- Tiene soporte para Spring y contenedores de inyección estándar (@Inject).
- Soportado en servidores Glassfish²⁴
- Es mucho más modular que los otros proyectos de JAX-RS²⁵.
- Cuenta con URI Constructor
- No requiere necesariamente contenedor de servlets.

Jersey²⁶ es más que la aplicación JAX-RS Referencia. Jersey ofrece su propia API que permite ampliar el conjunto de herramientas JAX-RS con las características y utilidades adicionales para simplificar aún más el servicio REST y desarrollo de clientes. Jersey también expone numerosas SPI de extensión para que los desarrolladores puedan extender Jersey para que se adapte mejor a sus necesidades_(Corporation, 2015).

¹⁹ http://es.wikipedia.org/wiki/Representational_State_Transfer

²⁰ <http://resteasy.jboss.org/>

²¹ <http://restlet.com/>

²² <http://cxf.apache.org/>

²³ <https://www.websocket.org/>

²⁴ <http://es.wikipedia.org/wiki/GlassFish>

²⁵ <http://es.wikipedia.org/wiki/JAX-RS>

²⁶ <https://jersey.java.net/>

4.1.3.3 JSON

JSON(JavaScript ObjectNotation - Notación de Objetos de JavaScript)²⁷ es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para las personas, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje *ideal para el intercambio de datos*_(Standard, 2015).

4.1.3.4 Gson

Gson²⁸ es una biblioteca implementada en Java que es utilizada para convertir objetos Java en su representación JSON. También se puede utilizar para convertir una cadena JSON a un objeto Java equivalente.

4.1.4 Diseño UML Backend

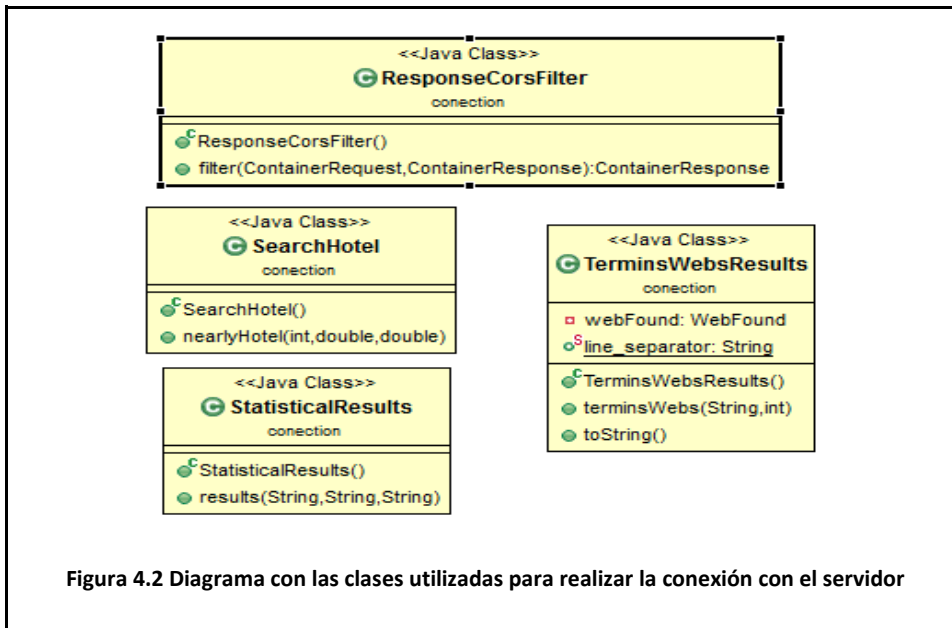
La implementación sigue el siguiente esquema UML²⁹:

Clases con las cuales establecemos la conexión al servidor

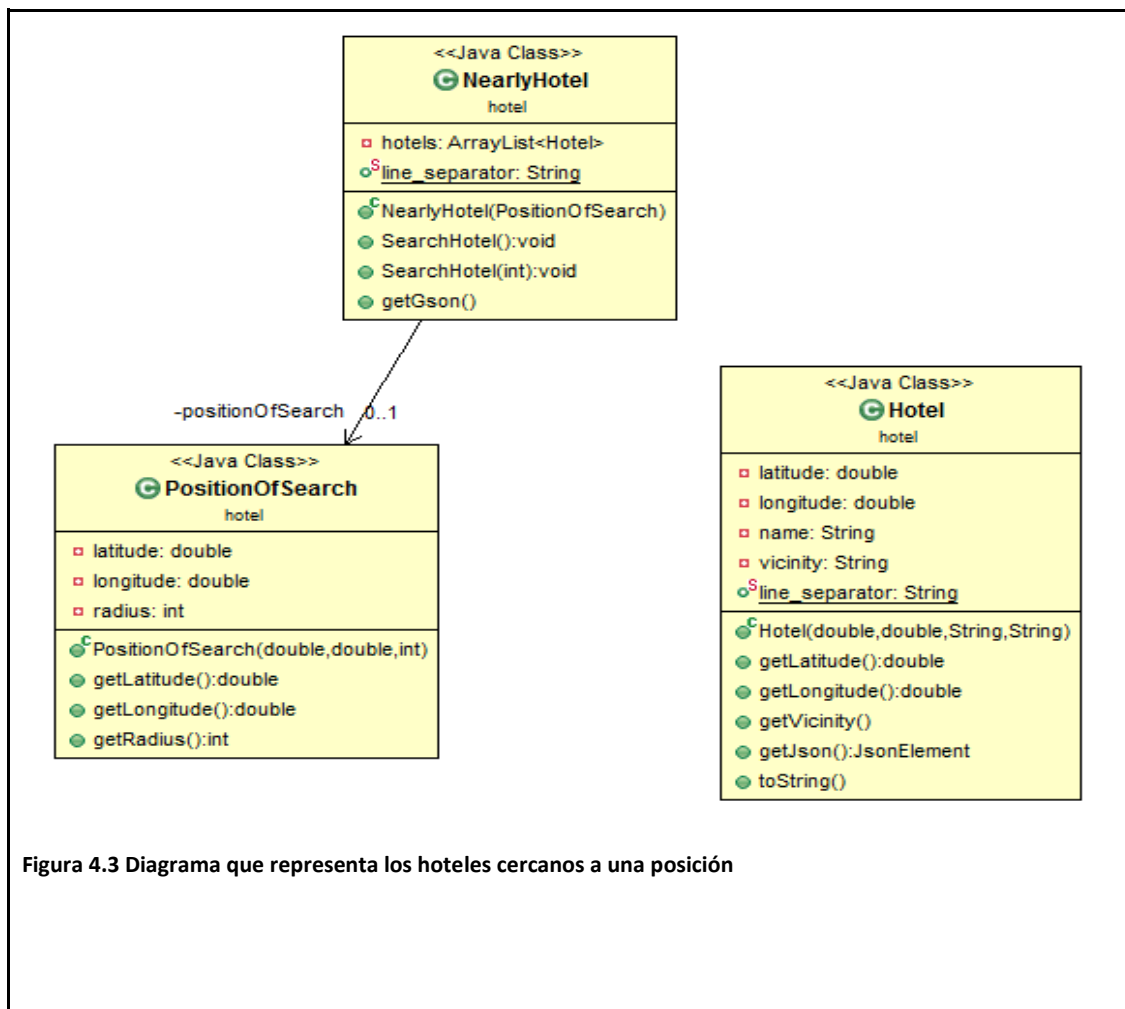
²⁷ <http://json.org/json-es.html>

²⁸ <https://code.google.com/p/google-gson>

²⁹ http://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado



Classes in which we represent the hotels near a position



Enumerados en la que implementamos nuestro diccionario a base de expresiones regulares.

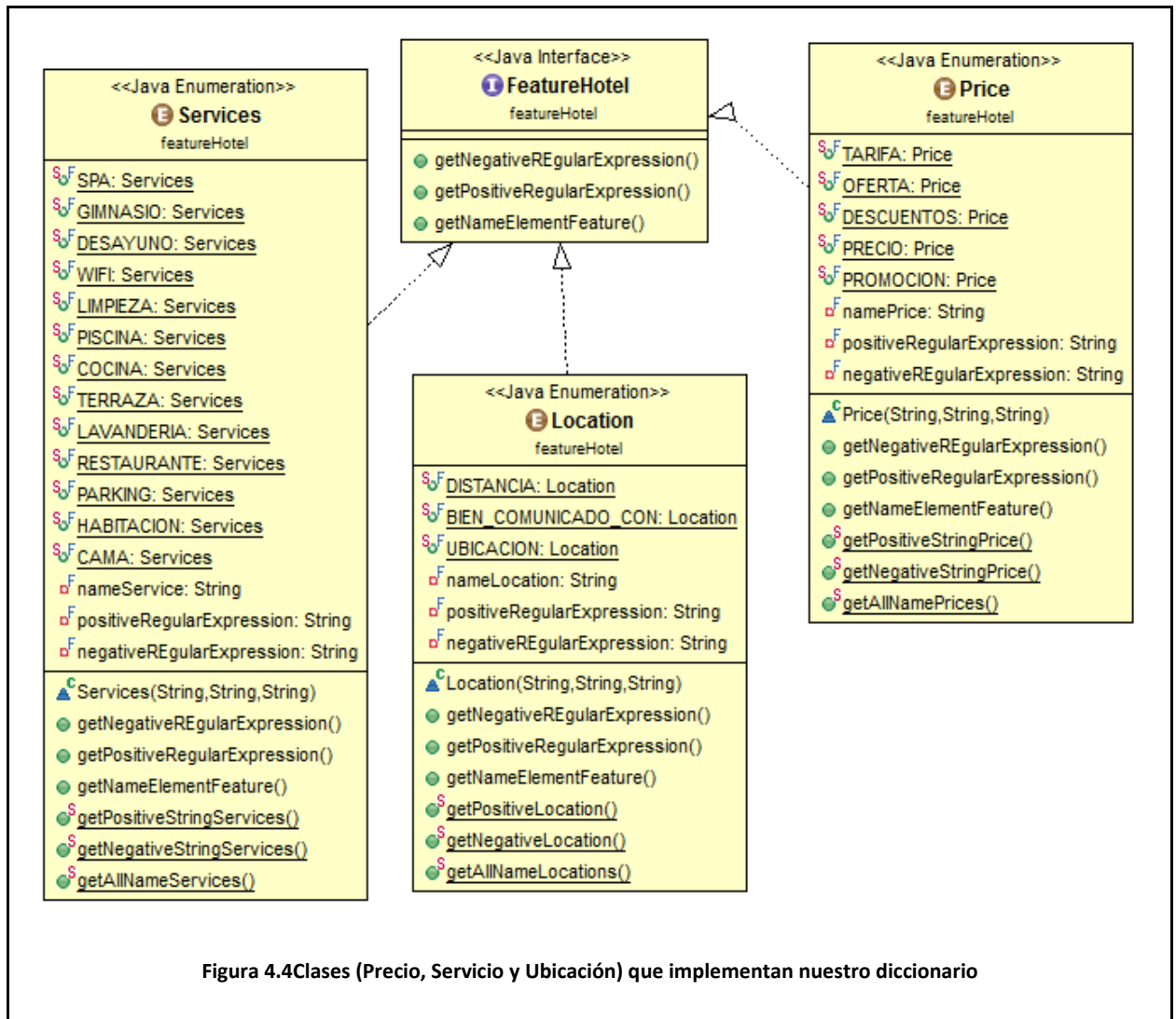


Figura 4.4 Clases (Precio, Servicio y Ubicación) que implementan nuestro diccionario

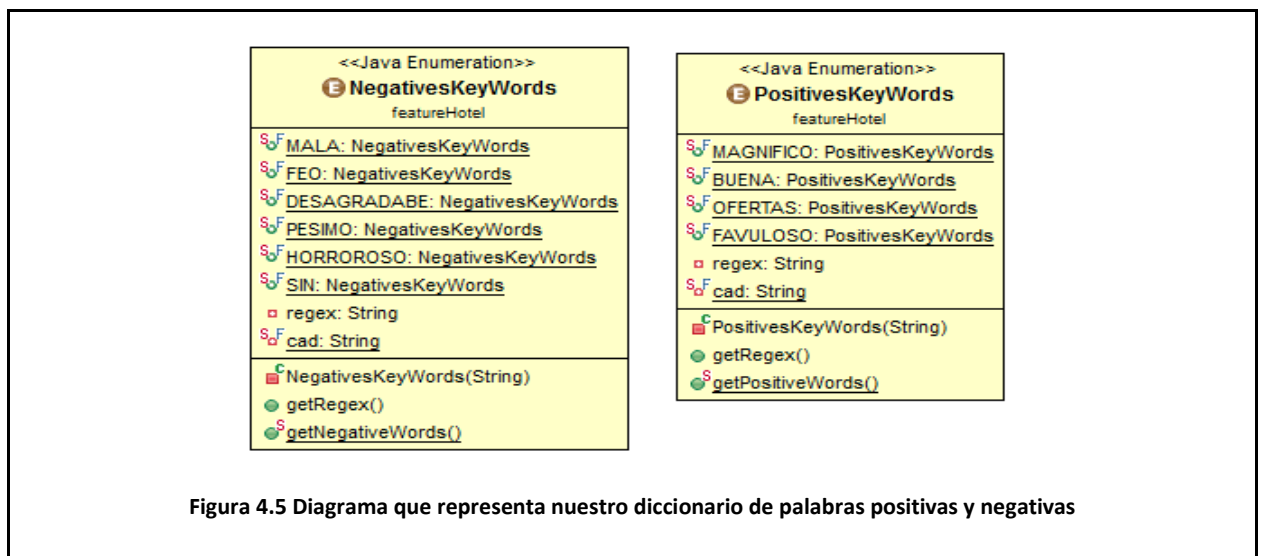


Figura 4.5 Diagrama que representa nuestro diccionario de palabras positivas y negativas

Clases que implementan métodos estáticos de los cuales hacemos uso para realizar conexiones, y procesamiento de String.

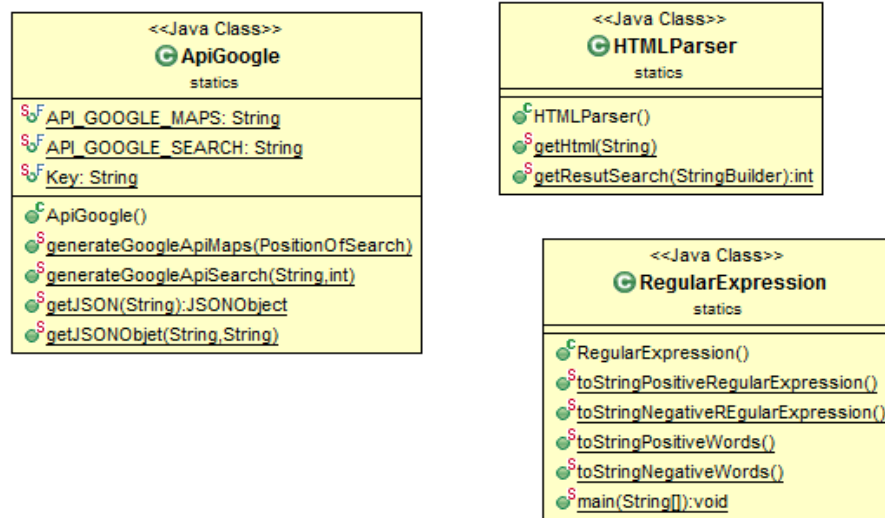


Figura 4.6 Diagrama con las clases auxiliares utilizadas

Clases que trabajan conjuntamente para realizar el análisis estadístico.

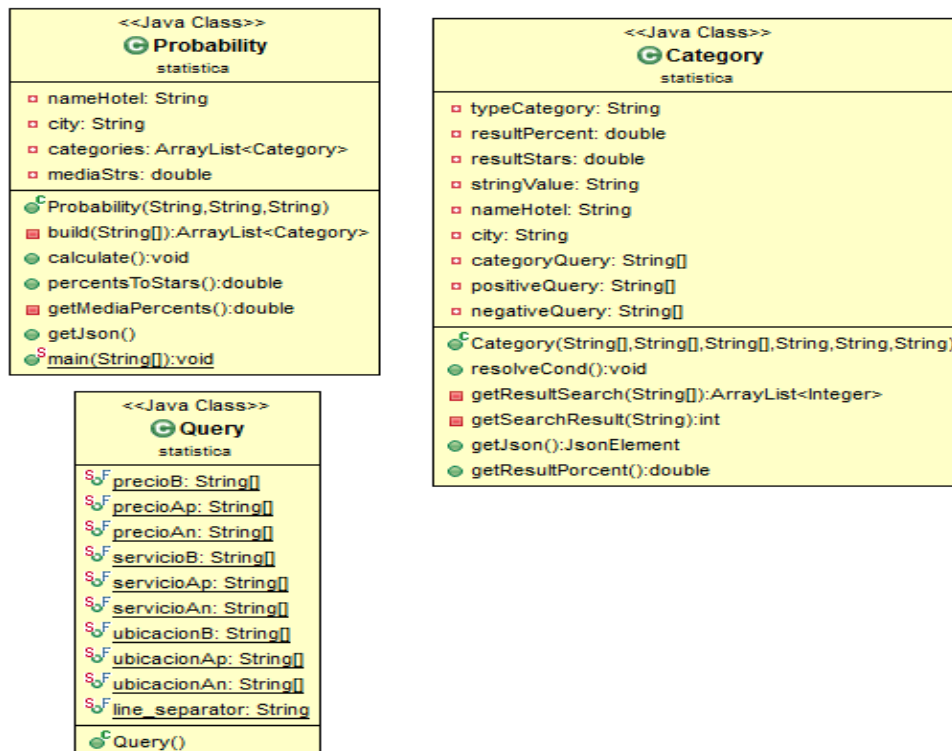
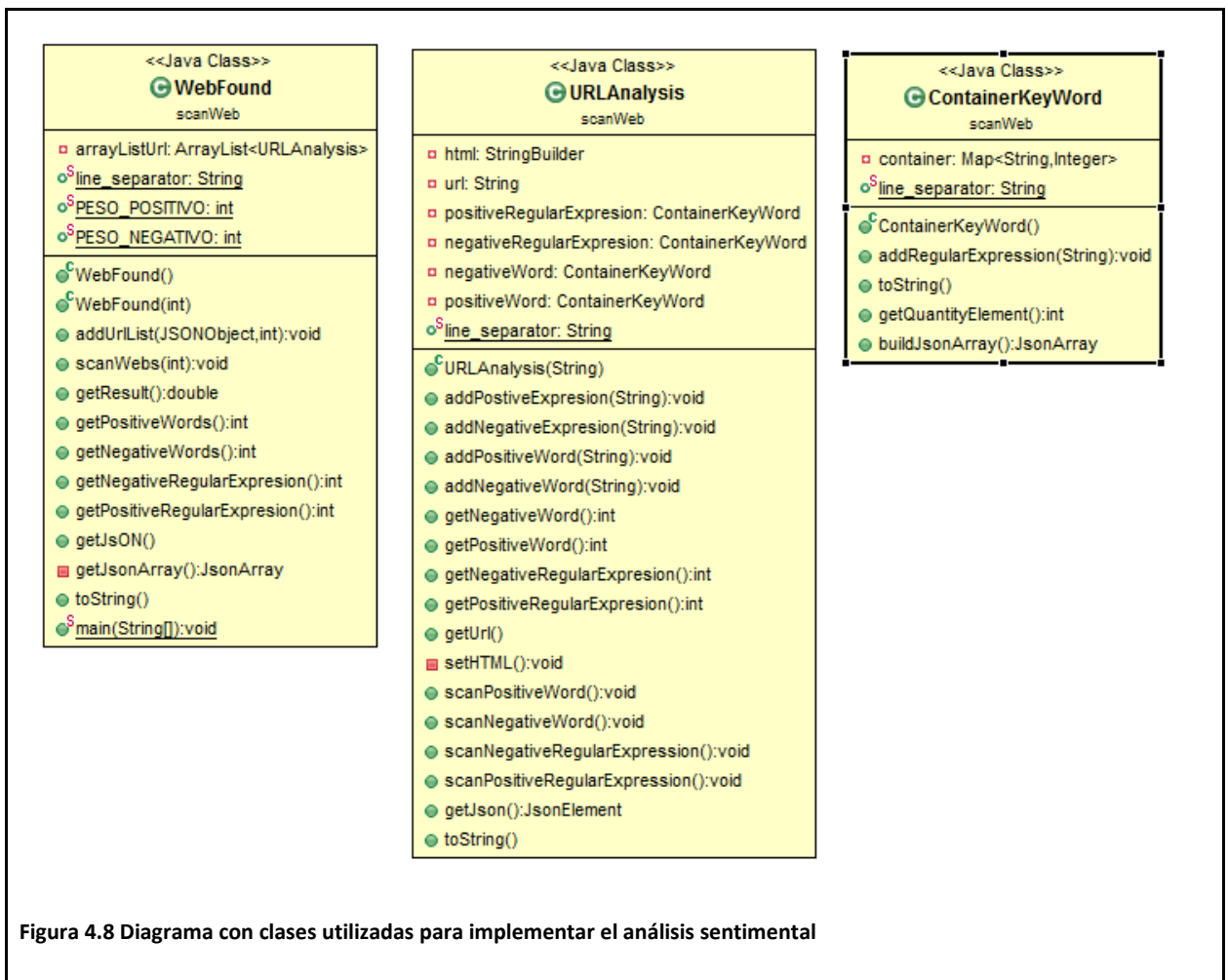


Figura 4.7 Diagrama con clases utilizadas para implementar el análisis estadístico

Clases que trabajan conjuntamente para obtener el análisis sentimental.



4.1.5 API (Interfaz de programación de aplicaciones)

Hemos diseñado e implementado una API que permite obtener la información generada por nuestro sistema mediante llamadas HTTP. De esta forma aplicaciones como SRH y Madrid Manía puedan utilizar esta información.

Las rutas de la API implementadas son:

Obtención de los 20 hoteles más próximos a una posición:

- valor_radio: indica el radio en el que se quiere buscar los hoteles, número entero
- valor_lat: indica la latitud del punto de búsqueda, número real (con 5 decimales)
- valor_long: Indica la longitud del punto de búsqueda, número real (con 5 decimales)

mlh.fdi.ucm.es/Backend/SearchHotel?radio=<valor_radio>&lat=<valor_lat>&lon=<valor_long>

p. ej. mlh.fdi.ucm.es/Backend/SearchHotel?radio=500&lat=40.435214&lon=-3.70235

Ej. Salida obtenida:

```
[
  {
    "name": "Tryp Madrid Chamberí-",
    "latitude": 40.438382,
    "longitude": -3.700857,
    "vicinity": "Calle José Abascal, 8, Madrid"
  },
  {
    "name": "Best Western Trafalgar Hotel",
    "latitude": 40.433962,
    "longitude": -3.701808,
    "vicinity": "Calle de Trafalgar, 35, Madrid"
  },
  {
    "name": "Sercotel Gran Hotel Conde Duque",
    "latitude": 40.430769,
    "longitude": -3.707741,
    "vicinity": "Plaza Conde Valle de Suchil, 5, Madrid"
  }
  ....,
]
```

Tabla 4.1 JSON con los hoteles cercanos

Obtención de las webs, palabras claves y expresiones regulares encontradas:

- query: Con esta consulta indicamos el contexto de las webs en la que queremos buscar
- cantida_webs: cantidad de webs que queremos explorar, máximo 57, número entero.

mlh.fdi.ucm.es/Backend/TermWebs?search=<query>&scan=<cantida_webs>

p. ej. mlh.fdi.ucm.es/Backend/TermWebs?search=Hotel%20Madrid%20Palace&scan=8

Ej. Salida obtenida:

```
{
  "stars": 3.5,
  "terminsWebs": [
    {
      "url": "http://www.westinpalacemadrid.com/en",
      "positiveRegularExpresion": 0,
      "negativeRegularExpresion": 0,
      "negativeWord": 0,
    }
  ]
}
```

```

"positiveWord": 4,
"positiveWordList": [
  {
    "type": "Gym / Fitness St",
    "value": 1
  },
  {
    "type": "Restaurants \u0026 Bars overv",
    "value": 1
  },
  {
    "type": "Spá 2009-2015 ",
    "value": 1
  },
  {
    "type": "Parking â,-39 per day ",
    "value": 1
  }
],
"negativeWordList": [],
"positiveRegularExpresionList": [],
"negativeRegularExpresionList": []
},
{
  "url": "http://www.westinpalacemadrid.com/",
  "positiveRegularExpresion": 0,
  "negativeRegularExpresion": 0,
  "negativeWord": 0,
  "positiveWord": 19,
  "positiveWordList": [
    {
      "type": "restaurante de altacoci",
      "value": 1
    }
  ],
  .....
},
  "negativeWordList": [],
  "positiveRegularExpresionList": [],
  "negativeRegularExpresionList": []
}
{....}
....
]
}

```

Tabla 4.2 JSON con los resultados webs evaluados

Obtención de resultados estadísticos en valor de porcentaje y estrella:

ciudad: ciudad en la que se encuentra el hotel.

- nombre_hotel: nombre del hotel a evaluar.
- lista _de categorías: categoría/as (precio,ubicación,servicio) concatenadas separadas por coma.

mlh.fdi.ucm.es/Backend/Results?city=<ciudad>&hotel=<nombre_hotel>&termino=<lista _de_categorias>

p.ej.

mlh.fdi.ucm.es/Backend/Results?city=Madrid&hotel=Palace&termino=precio,ubicacion,servicios

Ej. salida obtenida:

```
{
  "nameHotel": "Palace",
  "category": [
    {
      "type": "precio",
      "valuePercent": 95.22427440633246,
      "valueStars": 4.761213720316623,
      "valueString": "excelente"
    },
    {
      "type": "ubicacion",
      "valuePercent": 89.82758620689654,
      "valueStars": 4.491379310344827,
      "valueString": "excelente"
    },
    {
      "type": "servicios",
      "valuePercent": 28.500000000000004,
      "valueStars": 1.4250000000000003,
      "valueString": "malo"
    }
  ],
  "totalResult": 71.183953537743,
  "totalStars": 3.5591976768871496
}
```

Tabla 4.3 JSON con el resultado estadístico obtenido

Obtención del mejor hotel cercano:

Desarrollado específicamente para que la aplicación **Madrid Manía** lo utilice y pueda con ello brindar un mayor servicio al público.

Query: En ella estará la posición desde la cual se quiere encontrar el mejor hotel que nuestro Sistema Recomendador (SRH) devolverá.

- **Precondición:** indicar el punto de búsqueda de hoteles
 - **valor_latitud:** latitud (lat), punto en el cual se quiere buscar, número formado con cinco decimales
 - **valor_longitud:** longitud (lon), punto en el cual se quiere buscar, número formado con cinco decimales

`http://mlh.fdi.ucm.es/Backend/HSRgetBestHotel?lat=<valor_latitud>&lon=<valor_longitud>`

p. ej. `http://mlh.fdi.ucm.es/Backend/HSRgetBestHotel?lat=40.427758&lon=-3.696991`

Ej. Salida obtenida:

```
{  
  
  "stars": 4.743589743589744,  
  
  "nameHotel": "Hotel Petit Palace Santa Barbara"  
  
}
```

Tabla 4.4 JSON con el mejor hotel

Nota: El tiempo en obtener la respuesta está entre los 3 a 6 segundos, esto es debido al procesamiento que hace nuestro Backend con los datos.

4.2 Arquitectura Frontend

Hemos desarrollado una página web en el Frontend con diseño responsive que da al usuario una forma fácil e intuitiva de usarla y en pocos pasos obtenga respuestas a sus interacciones, además tecnologías como HTML5, Bootstrap mas otras que mencionaremos a continuación influyen en darle al usuario beneficios de usabilidad y movilidad.

La función básica del Fronted es servir como interfaz entre el usuario y el servidor a donde llegan las peticiones del cliente y se obtiene resultados del procesamiento de información que se ha solicitado para luego ser mostrados en nuestro servicio web.

Tenemos también la interacción y el manejo de eventos en el mapa, gracias al API de Google Maps con la que trabajamos para brindar al usuario facilidad de localización de las ciudades, hoteles y geo localización, con un botón creado y añadido desde el mismo Fronted para cumplir esta última funcionalidad.

Para representar la información que nos llega desde el Backend, comunicación vía JSON, precisamente parseamos dicha entrada con tecnologías como JQuery, JavaScript para el manejo y tratamiento de la información y presentarla de la mejor forma al usuario y esta sea distinguible, clara y rápida en ser mostrada.

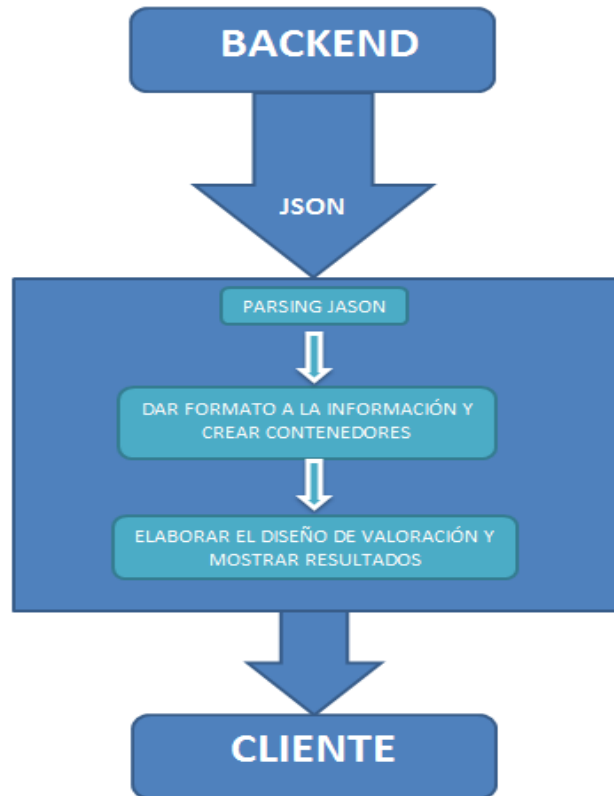


Figura 4.9 ArquitecturaFrontend

4.2.1 Tecnologías usadas para la implementación del Frontend

Hay bastante juego a la hora de manejar las tecnologías en el Frontend, por ello nuestro sistema lleva varias de ellas mezaclando los beneficios, de cada una de acuerdo a las necesidades que hemos tenido, sobre todo para manejar eventos en la web, animaciones, tipos de datos, valoraciones, componentes y marcadores en el mapa, etc. A Continuación Explicamos estas tecnologías con un detalle general.

4.2.1.1 HTML5

HTML5 (*HyperTextMarkupLanguage*, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como *HTML5* y una variante XHTML

conocida como sintaxis *XHTML5* que deberá ser servida como XML_(Wikipedia, Wikipedia, 2015).

Al crear una web el lenguaje básico para desarrollarla es HTML, pero con la intención de crear un sitio más profesional, completo y moderno hemos usado la última versión. HTML5 nos ayuda a cumplir los beneficios anteriores además de mejorar en usabilidad, importante para el buen manejo de la página por el usuario, para ello hemos usado las nuevas etiquetas que esta versión dispone y también sus diferentes atributos, para luego ser complementados por funciones en JavaScript o estilizar con CSS.

4.2.1.2 CSS3

Hoja de estilo en cascada o CSS (siglas en inglés de *cascadingstylesheets*) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML² (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores_(Wikipedia, Wikipedia, 2015).

Darle al usuario una presentación bien diseñada es el trabajo de esta tecnología, CSS son las hojas de estilo relacionadas al HTML, cumple con la tipografía, posicionamiento de elementos, bordes y hasta con ciertas animaciones que hacen atractiva a nuestra web y bastante amigable para que contribuyan a la memorabilidad en la parte del usuario y además contiene conceptos de consistencia y buenas prácticas de diseño de aplicaciones web.

4.2.1.3 JavaScript

JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico_(Wikipedia, Wikipedia, 2015).

Al ejecutar las acciones de los botones en la página web, estas acciones llaman a funciones que están desarrolladas en JavaScript casi la mayoría del código en el Frontend es de este tipo de tecnología, da facilidad de programación es quien por ejemplo muestra los hoteles cercanos haciendo llamadas al backend con parámetros como la query definida para cada servicio que más las APIs de Google modelamos la información de entrada y presentamos sus resultados.

4.2.1.4 JQuery

jQuery³⁰ es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCampNYC. jQuery es la biblioteca de JavaScript más utilizada_(Wikipedia, Wikipedia, 2015).

³⁰ <https://jquery.com/>

Este lenguaje nos ha representado un poco de trabajo para controlarla pero una vez manejas el API todo va más sencillo, hemos utilizado JQuery en casi toda la parte de eventos y cambios de clases para la animación o cambios de estilos, las funciones de parseo de la información que nos envía el Backend es tratada también bajo esta tecnología y también ha sido usada para control de animación de distintos componentes en la web y creación de nuevos elementos automáticamente.

4.2.1.5 Bootstrap

TwitterBootstrap³¹ es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.(Wikipedia, Wikipedia, 2015).

Usamos este framework por el apoyo grande que da a la hora de diseñar una web, la distribución es mejor controlada gracias a la combinación de esta tecnología con HTML5, además cabe mencionar las ventajas respecto a su diseño responsive el cual es clave para que la web esté presente en cualquier dispositivo de mesa o móvil y su visualización sea usable y se maneje de forma eficaz.

4.2.1.6 Material

Material design³² es un concepto, una filosofía, unas pautas enfocadas al diseño utilizado en Android, pero también en la web y en cualquier plataforma, desarrollado por Google y anunciado en la conferencia Google I/O celebrada el 25 de junio de 2014. Ampliando la interfaz de tarjetas vista por primera vez en Google Now, Material se trata de un diseño más limpio, en el que predominan animaciones y transiciones de respuesta, el relleno y los efectos de profundidad tales como la iluminación y las sombras.(Wikipedia, Wikipedia, 2015).

Aprovechamos el auge de Material design para el diseño de la página web, en cuanto a dar profundidad a un estilo plano que nos da Bootstrap, esto estiliza los botones y los paneles, dándoles mejores acabados como efectos en las esquinas y sombras que indiquen un grado de relieve o cuando se use el mouse realizar alguna diferenciación en la animación.

4.2.2 Diseño UML Frontend

La implementación del Frontend se puede seguir mediante los siguientes ejemplos de casos de uso, que muestran el funcionamiento y trabajo que realizamos en interfaz con el usuario, las

³¹ <http://getbootstrap.com/>

³² <http://www.google.com/design/>

peticiones de información hacia el Backend, los datos que procesamos y cómo se muestran los resultados al usuario.

Ubicación en el mapa

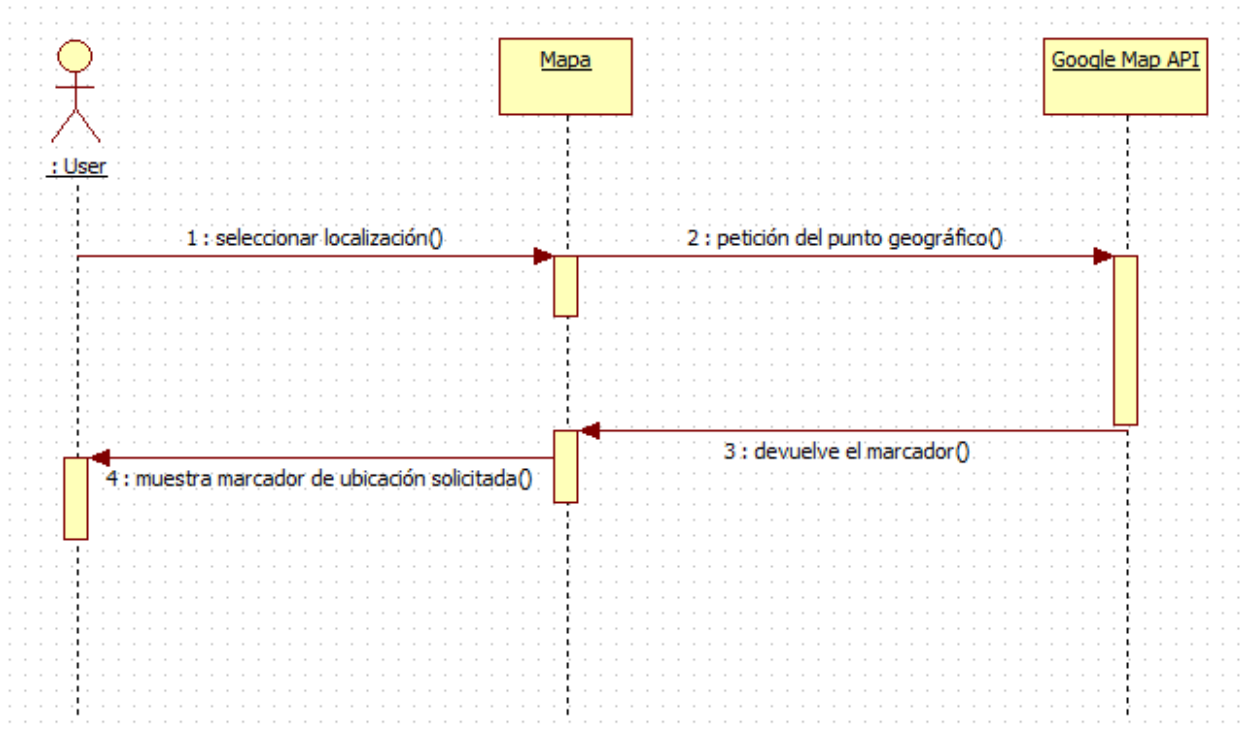


Figura 4.10 Diagrama de secuencia UML para obtener la ubicación en el mapa

Acción	Descripción
1 seleccionar localización	Tenemos tres formas distintas para ubicar una posición: <ol style="list-style-type: none"> 1. buscar una ciudad, esto nos ubicará en el centro urbano de la ciudad elegida. 2. hacer click en el mapa, al hacerlo inmediatamente señalamos dónde está nuestro sitio. 3. geo localización, una vez que hemos permitimos la activación del GPS automáticamente el sistema sabrá la posición actual.
2 petición del punto geográfico	Llamaremos a las funciones del API de Google Maps con las que se trabaja en el Frontend para obtener un punto en el mapa.
3 devuelve el marcador	Obtenemos la localización en un marcador mostrado en el mapa.
4 mostrar marcador de la ubicación	Pintar un marcador personalizado en el mapa para resaltar

solicitada	el punto devuelto por el API de Google Maps.
------------	--

Tabla 4.5 Descripción de la ubicación en el mapa

Petición de hoteles cercanos

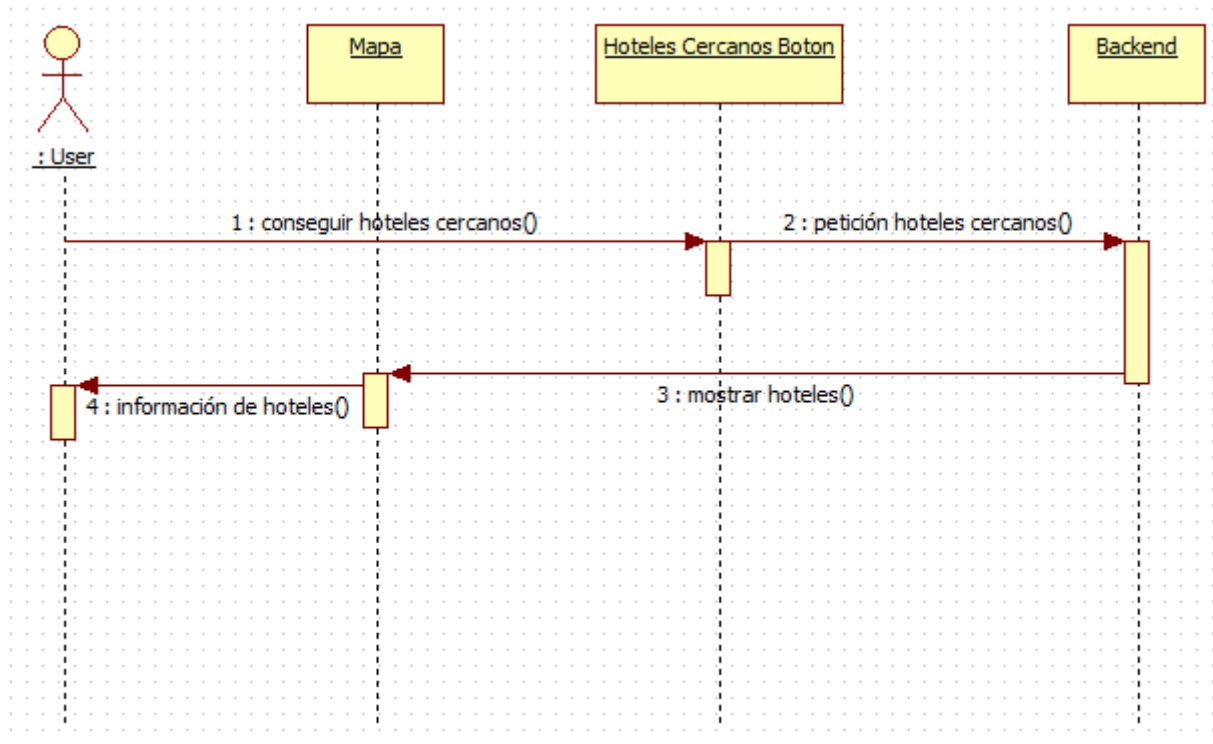


Figura 4.11 Diagrama de secuencia UML de la petición de hoteles cercanos

Acción	Descripción
1 conseguir hoteles cercanos	Interacción del usuario con el botón de acción de hoteles cercanos.
2 petición de hoteles cercanos	Conectamos con el Backend solicitando los hoteles cercanos y este hará el procesamiento necesario comunicándose con el API de Google Place.
3 mostrar hoteles	Una vez se haya procesado la información devuelta por el Backend, los resultados se envían nuevamente al mapa mostrando iconos personalizados tarea del Frontend.
4 información de hoteles	Además de mostrar los hoteles cercanos con diseño propio el Frontend añade la funcionalidad de poder mirar información de los hoteles con un solo click en el marcador y autocompletar el

	campo de nombre del hotel en el formulario para futuros análisis.
--	---

Tabla 4.6 Descripción de la petición de hoteles

Petición del servicio de recomendación estadística

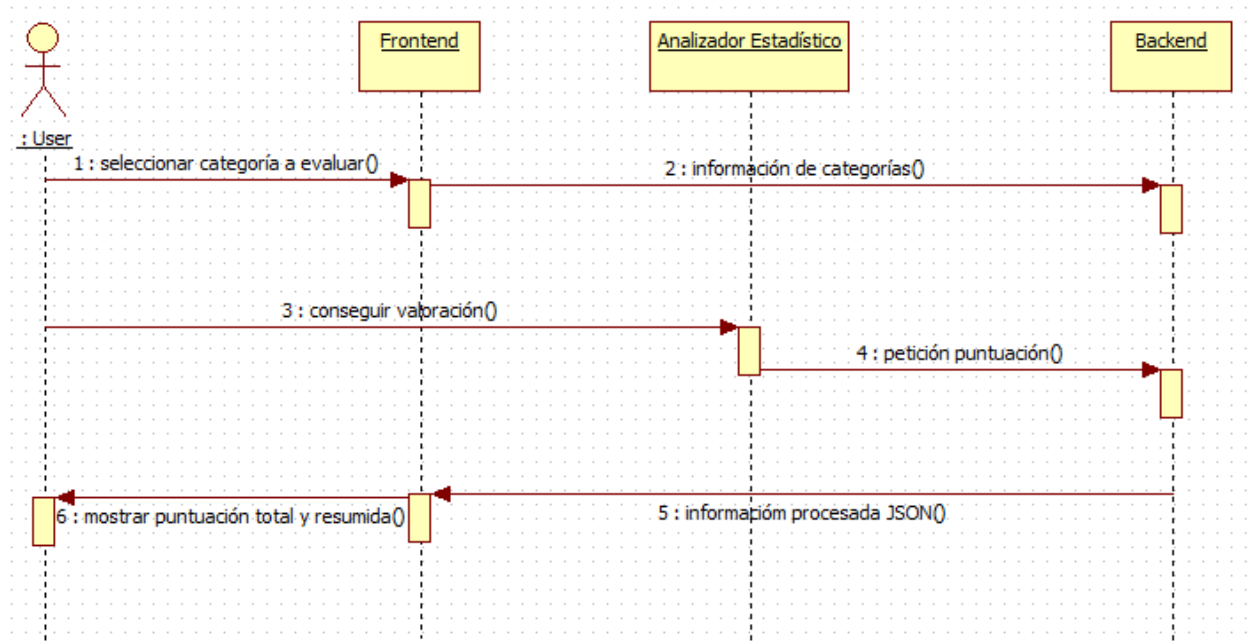


Figura 4.12 Diagrama de secuencia UML de la petición del análisis estadístico

Acción	Descripción
1 seleccionar categorías a evaluar	Una vez seleccionado el hotel que el usuario desea evaluar, se puede elegir entre las características de precio, ubicación y servicios o las tres al mismo tiempo dependiendo de las necesidades o el perfil del usuario.
2 información de categorías	La selección anterior servirá para el cálculo total y parcial del hotel y se informa al Backend de ello.
3 conseguir valoración	Después de seleccionar las categorías el usuario pedirá la evaluación estadística.
4 petición de puntuación	Una vez completado los pasos anteriores se genera la petición al Backend con la información a consultar.
5 información procesada	El Backend genera los resultados después de entrar a nuestro algoritmo del cálculo estadístico implementado y lo devuelve en

	un JSON
6 mostrar puntuación total y resumida	En el Frontend se parsea el JSON y genera componentes y diseño para la presentación de los resultados.

Tabla 4.7 Descripción de la petición del análisis estadístico

Petición del servicio de recomendación sentimental

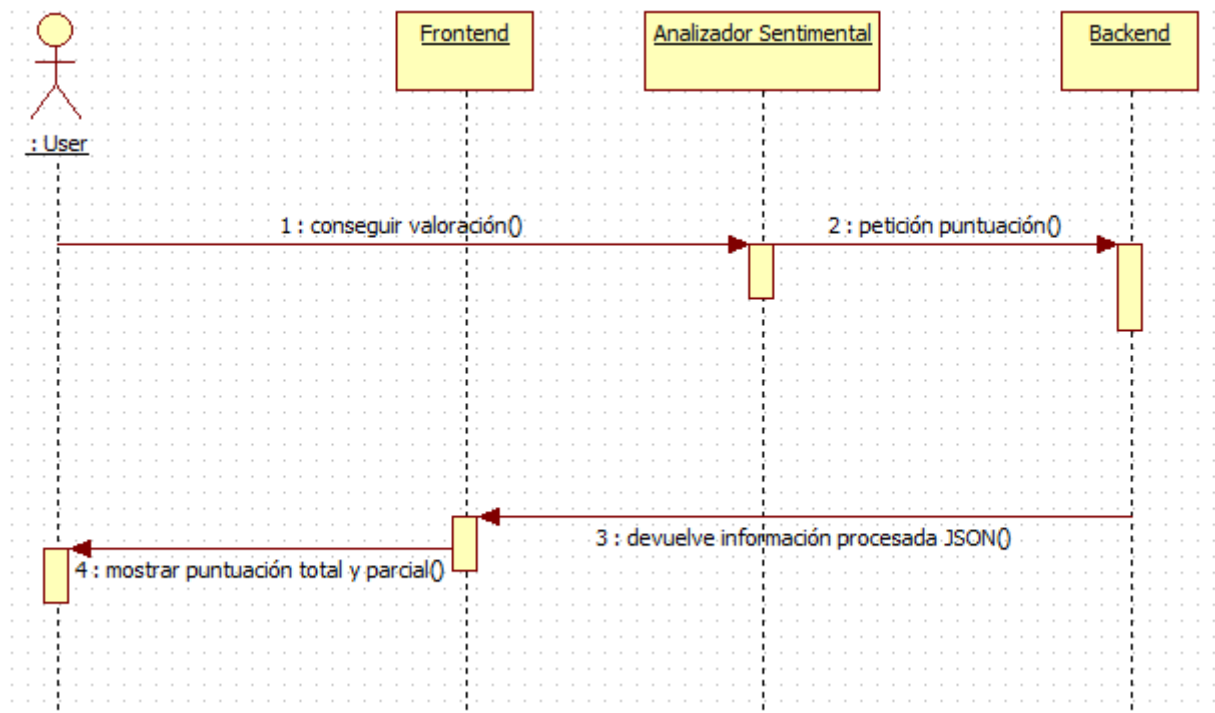


Tabla 4.8 Diagrama de secuencia UML del análisis sentimental

Acción	Descripción
1 conseguir valoración	Cuando ya se haya seleccionado el hotel que el usuario desea evaluar, se pide la evaluación sentimental.
2 petición puntuación	Se pide desde el Frontend el análisis sentimental al Backend el cual hará el trabajo de poner en proceso el escaneo de las páginas web y realizar los cálculos que se han explicado en este mismo capítulo en la arquitectura del Backend.
3 devuelve información	Luego del trabajo del Backend se genera un JSON que contendrá

procesada JSON	la información resultante.
4 mostrar puntuación total y parcial	Lo que queda es dar una presentación a la información devuelta esta vez mostraremos al usuario puntuación total y en detalle de las páginas consultadas.

Tabla 4.9 Descripción de la secuencia del análisis sentimental

5 Descripción Funcional

Y por fin llegamos a la parte donde el usuario interactúa con nuestro sistema por medio de nuestra página web, a continuación describiremos las tareas más relevantes y comunes que el usuario normalmente haría, para esto tomaremos en cuenta los casos de uso vistos en el capítulo anterior de la parte del Frontend³³, representaremos cada acción con una captura de pantalla y la describiremos, la entrada, el proceso y la salida brevemente para cada evento realizado. Cabe destacar que en este punto de la aplicación todo depende del usuario, siendo este quien tiene el control de todo y probamos a ver si el diseño es tal, que el cliente no tenga problemas de usabilidad y el sistema sea cómodo de manejar y recordar.

Los siguientes pasos representan toda una iteración de un usuario en su proceso de llegar a completar el circuito de recomendación de un hotel elegido.

5.1 Definir posición

Como se ha mencionado en ocasiones anteriores el usuario puede señalar la ubicación desde donde pretende buscar hoteles cercanos, para ello en la tabla de abajo definimos tres numerales los cuales serán descritos cada uno.

³³ <http://mlh.fdi.ucm.es/SRH/>



Figura 5.1 Definir posición ciudad

Acción	Descripción
1 Selección de ciudad	El usuario puede definir la posición en el centro de la ciudad que haya elegido dentro de España
2 Mostrar marcador	Una vez elegida la ciudad se mostrará un marcador con un cuadro de información que lleva el nombre de la ciudad y su escudo que la representa, este último si damos click sobre él, conduce a Wikipedia donde se tiene información de la ciudad.

Tabla 5.1 Definir posición ciudad

Aún quedan dos formas de ubicarnos en el mapa, para ponerlo claro otra imagen y tabla explicativa.



Figura 5.2 Definir posición mapa y geolocalización

Acción	Descripción
1 Ud. está aquí	El famoso Ud. está aquí, lo podemos obtener con un solo click sobre el mapa, podemos explorar el mapa si fueran las necesidades del usuario hasta dar con más precisión en un punto y desde ahí invocar a los hoteles cercanos para ser evaluados.
2 Geo localizador	Activar el GPS a muchos nos saca de apuros y es muy útil, esta facilidad la aprovechamos para dar opciones y actualidad a nuestro servicio.

Tabla 5.2 Definir posición mapa y geolocalización

5.2 Buscar hoteles cercanos

Una vez hemos seleccionado el punto de partida para la búsqueda de hoteles cercanos seguiremos con lo siguiente.



Figura 5.3 Hoteles cercanos

Acción	Descripción
1 Activar el botón hoteles cercanos	Al empezar con la búsqueda de hoteles cercanos, daremos click en botón de hoteles cercanos para enviar las peticiones correspondientes al Backend y que se nos regrese esta información al mapa.
2 Visualización en el mapa	Luego de la acción del evento, podemos fijarnos en todos los hoteles que aparecen en el mapa, de ellos el usuario elegirá cualquiera que prefiera y pueda finalmente obtener una valoración tanto estadística como sentimental.

Tabla 5.3 Hoteles cercanos

Seleccionar hotel a analizar

Ya que se han encontrado varios hoteles cerca a la posición deseada el usuario seleccionará uno de los hoteles de interés y se notará los siguientes eventos.

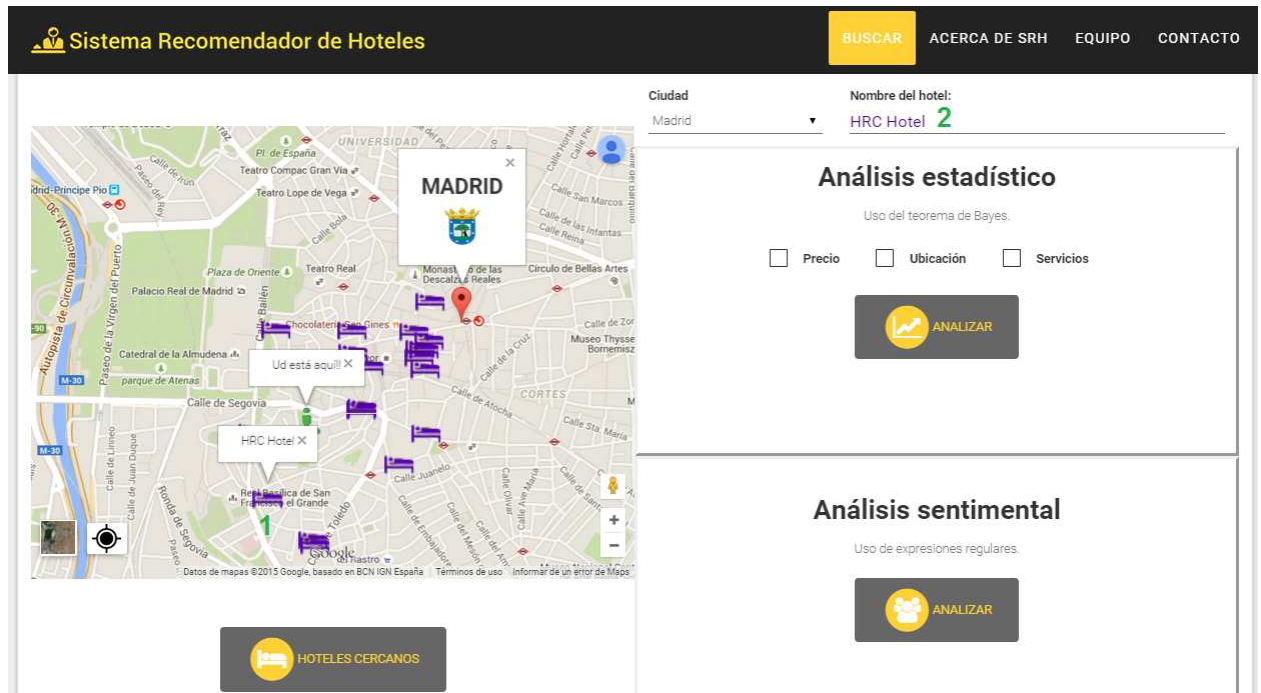


Figura 5.4 Selección de hotel

Acción	Descripción
1 Selección de hotel	Con un click sobre el hotel al que se quiere referenciar, se mostrará un cuadro de mensaje con el nombre del hotel que sale del marcador del hotel personalizado.
2 Autocompletado nombre del hotel	Automáticamente después de haber seleccionado el hotel con el mismo click de antes se autocompleta el nombre del hotel en el formulario, luciendo un estilo consistente con el icono del marcador del hotel y esté listo para su valoración.

Tabla 5.4 Selección de hotel

5.3 Analizador Estadístico y Sentimental

Finalmente queda realizar ya la evaluación al hotel, para ello tenemos dos analizadores como hemos mencionado en otros apartados y comenzaremos como sigue la tabla más abajo.

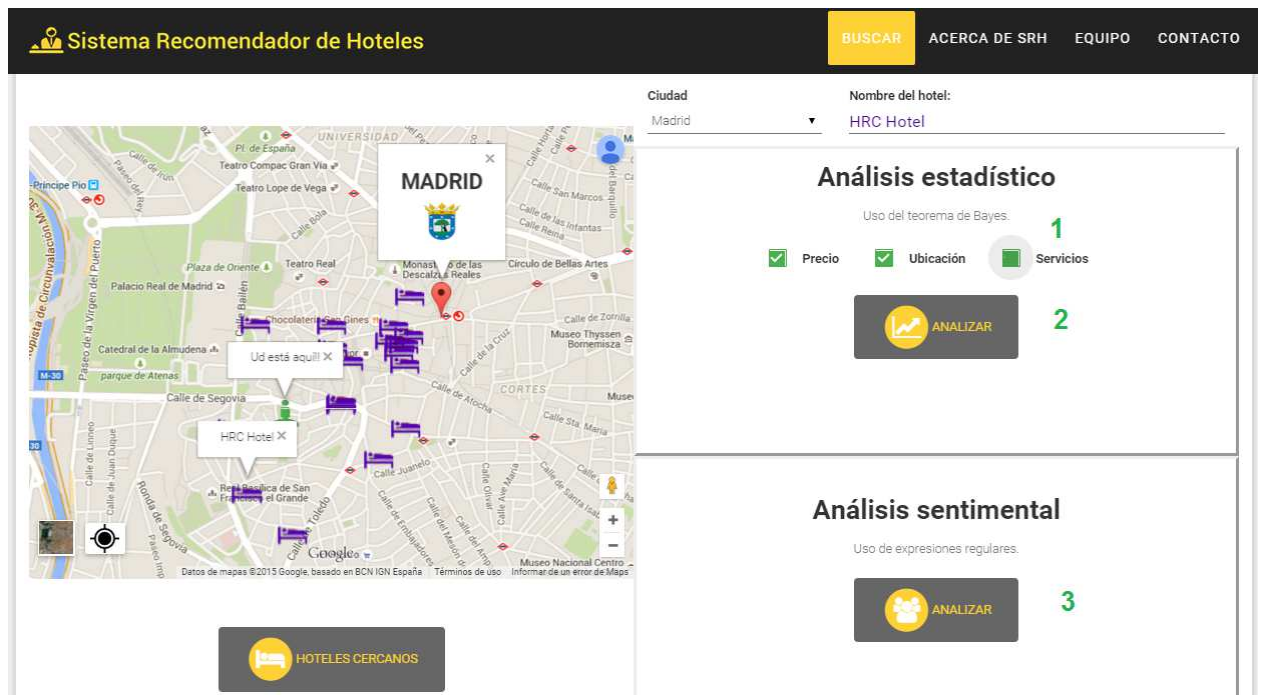


Figura 5.5 Analizador estadístico y sentimental

Acción	Descripción
1 Selección de categorías	Para un Análisis Estadístico se debe por lo menos seleccionar una de las tres características que aparecen en los checkboxes, las mismas que se relacionan de acuerdo a los perfiles de usuario, cabe destacar que la activación de una categoría u otra influirá directamente en la puntuación total y parcial de un hotel.
2 Analizador estadístico	Hecha la selección anterior, el usuario procederá a pedir los resultados y estos se mostrarán como se nota en la figura aquí abajo.
3 Analizador sentimental	También se puede hacer a la vez que el estadístico, pero se tardará mucho más, porque deberá ir escaneado todas las webs en búsqueda de palabras que mencionan al hotel.

Tabla 5.5 Analizador estadístico y sentimental

Resultados del Análisis Estadístico, cuenta por una parte con la valoración total en formato más grande y con estrellas dinámicamente asignadas, además de mostrar información de resultados parciales, de acuerdo a las categorías escogidas anteriormente.



Figura 5.6 Resultado analizador estadístico

Resultados del Analizador Sentimental, vemos la animación para darle feedback al usuario de espera mientras la tarea está siendo ejecutada.

Análisis sentimental

Uso de expresiones regulares



Figura 5.7 Animación de espera análisis sentimental

Una vez cumplido toda la comprobación de expresiones referentes al hotel en juego, vemos una puntuación total de igual formato que la anterior, es decir, se luce un número acompañado de estrellas para señalar su puntuación total, y por otra parte tendremos las webs que hacen la conexión con el hotel y serán los sitios en Internet donde se hable del hotel en cuestión.

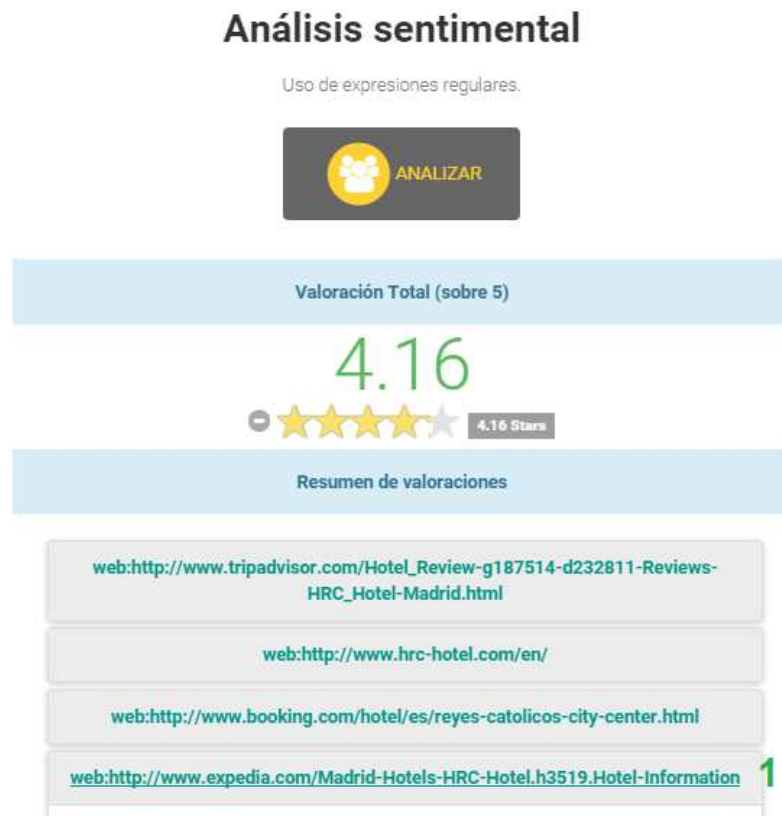


Figura 5.8 Resultado análisis sentimental

Acción	Descripción
1 Selección de una web	Del conjunto de webs se puede ir viendo una a una y la información que veremos en ellas será las listas de expresiones o palabras positivas y lista de expresiones o palabras negativas, si fuera el caso, cada palabra llevará un peso que corresponde al número de veces que se habla en el mismo sitio de un hotel específico, y se muestra el ejemplo a continuación.

Tabla 5.6 Selección web del resultados sentimental

RESULTADOS	
Lista de palabras positivas (+)	
Ubicación Este hotel	1
Desayuno disponible (1
cama doble o 2 in	1
desayuno continental:	1
precio aproximado).	1
Oferta hoteles Viaj	1
Wifi gratuita Dis	1
precio Ayuda Atenci	1
Wifi gratuita Apa	1
cerca del hotel Op	1
precio Garantía de	1
distancia Cruceros Esq	1
precio de Expedia S	1
Limpieza de habitacio	1
wifi gratis y min.	1
terrazza Ocupación má	1
Wifi gratis Camas	1
wifi de alta velo	1
desayuno continental	1
wifi en las zonas	1
ofertas: Quiénes somo	1
Wifi gratuita Con	1
Wifi gratuita Acc	1
Lista de palabras negativas (-)	
sin asistencia y una	1
sin asistencia (de p	2
sin asistencia: 20 E	1
Lista de expresiones regulares positivas (+)	
wifi gratis	1
Wifi gratis	1

Tabla 5.7 Resultado detallado del análisis sentimental

Como vemos en pocos clicks el usuario ha tenido una recomendación estadística y sentimental, el manejo de nuestra web presta los servicios de búsquedas, autocompletación y recomendación de una forma bastante fácil e intuitiva como se nota en los pasos que hemos detallado antes.

6 Test

6.1 Conexión

Una vez implementada toda la aplicación del lado del servidor, hemos realizado la comprobación de su buen funcionamiento, para ello seleccionamos 160 hoteles de 16 ciudades diferentes de España, con 10 hoteles por ciudad agrupados por nivel de puntuación, cantidad de estrellas, dadas por los buscadores de TripAdvisor y Trivago. Hemos almacenados los mismos en un fichero que más tarde ha de ser leído por una aplicación (test) encargada de llevar a cabo la ejecución tanto del Análisis Estadístico como la del Sentimental. Dicho análisis no fue nada fácil ya que Google nos limitaba en sus servicios a la hora de realizar determinadas consultas a sus servidores, esto nos ha obligado a buscar una solución la cual nos permitiera evadir tal restricción, fue aquí donde nos topamos con la red **TOR**³⁴, instalamos Tor en una máquina (Barbero, 2012), para verificar su funcionamiento y después de unos cuantos pasos pudimos conectarnos como cliente.

```
usuario@debian:~$ wget -q -O - checkip.dyndns.org | sed -e 's/.*Current IP Address: //' -e 's/<.*$//'  
188.138.17.15  
usuario@debian:~$ sudo service tor restart  
[sudo] password for usuario:  
[ ok ] Stopping tor daemon...done.  
[ ok ] Starting tor daemon...done.  
usuario@debian:~$ wget -q -O - checkip.dyndns.org | sed -e 's/.*Current IP Address: //' -e 's/<.*$//'  
171.25.193.78
```

Figura 6.1 Conexión con TOR

Como se puede observar en la imagen nuestra IP cambiaba, pero esto no fue suficiente ya que por algún motivo Google aun nos limitaba el servicio lanzándonos el siguiente mensaje de error.

```
Server returned HTTP response code: 503 for URL: http://ipv4.google.com/sorry/In  
dexRedirect?continue=http://www.google.es/search%3Fq%3DMadrid%26gws_rd%3Dssl&q=C  
GMSBKsZwU4Ym6OoqwUiGQDxp4NLDfJ39mkfgkMCHG1rjrznKhEzLNc
```

Figura 6.2 Erro 503

El cual nos dirigía a la siguiente página

³⁴ <https://www.torproject.org/>

Para continuar, introduce los caracteres que
aparecen a continuación:



Acerca de esta página

Nuestros sistemas han detectado tráfico inusual procedente de tu red de ordenadores. En esta página se comprueba si eres tú quien envía las solicitudes en lugar de un robot. [¿A qué se debe esto?](#)

Dirección IP: 81.60.153.230
Hora: 2015-05-30T19:30:12Z
URL: http://www.google.es/search?q=Madrid&gws_rd=ssl

Figura 6.3 Redirección de Google

Así que tuvimos que buscar otra solución, pero mientras tanto para tener algunos resultados, desarrollamos una simple aplicación instalada en el servidor la cual realizaba cada dos horas un total de 63 peticiones, número máximo antes de obtener el primer mensaje de error de Google, pero no nos quedamos nada conforme ya que necesitábamos un total de 27×160 peticiones para el análisis estadístico, más 50×160 para el análisis sentimental, esto hacía un total de 8.432 peticiones, sacando cuentas tendríamos que esperar un total de 133 horas, 6 días aproximadamente, trabajo poco viable.

Finalmente optamos por conectarnos a Google a través de un servidor proxy mediante la aplicación UltraSurf³⁵, es aquí donde dimos con la solución, el trabajo fue sencillo, simplemente teníamos que realizar la conexión a través de un socket y escuchar por el puerto 9666.

³⁵ <http://en.wikipedia.org/wiki/Ultasurf>

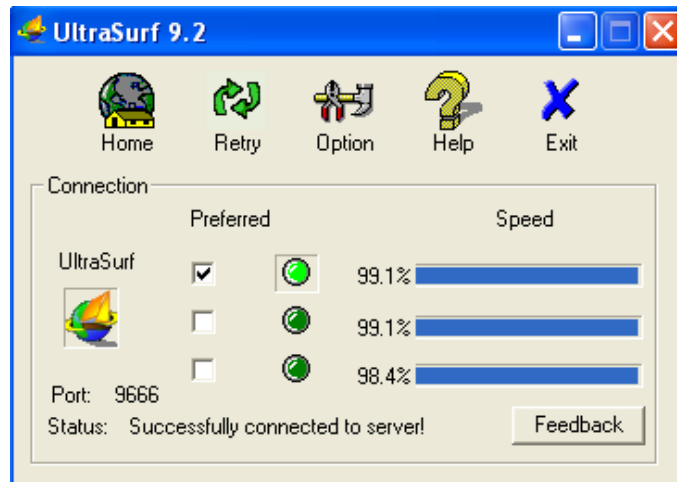


Figura 6.4 Aplicación UltraSurf

```
java.net.Proxy proxy = new java.net.Proxy (java.net.Proxy.Type.HTTP, new InetSocketAddress("127.0.0.1", 9666));
try {
    url = new URL(string);
    conn = (URLConnection) url.openConnection(proxy);
    conn.addRequestProperty("User-Agent", "Mozilla/4.76");
    conn.setRequestMethod("GET");
```

Figura 6.5 Conexión a un proxy

```
public static StringBuffer getHtml(String string) {
    URL url = null; // The URL to read
    HttpURLConnection conn; // The actual connection to the web page
    BufferedReader rd; // Used to read results from the web page
    String line; // An individual line of the web page HTML
    StringBuffer result = new StringBuffer();// A long string containing all the HTML
    java.net.Proxy proxy = new java.net.Proxy (java.net.Proxy.Type.HTTP, new InetSocketAddress("127.0.0.1", 9666));
    try {
        url = new URL(string);
        conn = (URLConnection) url.openConnection(proxy);
        conn.addRequestProperty("User-Agent", "Mozilla/4.76");
        conn.setRequestMethod("GET");
        //Thread.sleep(1000);
        rd = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        while ((line = rd.readLine()) != null) {
            result.append(line);
        }
        rd.close();
    } catch (Exception e) {
        System.err.println("Connection fail");
        e.printStackTrace();
    }
    return result;
}
```

Figura 6.6 Método que obtiene el HTML de Google

De esa manera, logramos obtener los resultados de Google, el siguiente paso fue analizarlos, para ello almacenamos los resultados en un fichero, fichero que más tarde con la ayuda de la librería Poi desarrollada en Java sería pasado a una hoja de cálculo Excel en la cual aplicaríamos nuestros algoritmos implementados para finalmente obtener gráficas que nos ayudarían a comparar nuestros valores con los de los potentes sistemas de búsqueda de hoteles.

6.2 Resultados

Con los datos obtenidos más los disponibles en TripAdvisor y Trivago calculamos el MAE (error absoluto medio) el cual nos muestra la cantidad de error que ha cometido nuestro sistema con respecto a los mencionados, cabe destacar que los valores que tiendan a cero serán los mejores en cuanto a que el error es mínimo y nuestros resultados se parecen más a valores reales.

6.2.1 Estudio del análisis estadístico

Tabla que contiene la media del MAE por ciudades³⁶.

Ciudad	Media MAE
Alicante	1,8651404897
Barcelona	1,6461272897
Bilbao	1,2290894886
Burgos	1,1776858452
Donostia San Sebastián	1,3498304307
Girona	0,9898301973
Granada	1,2889542969
Madrid	1,5837431425
Palma de Mallorca	1,2258486963

³⁶ <http://mlh.fdi.ucm.es/SRH/testEstadistico.pdf>

Salamanca	1,5843694312
Santa Cruz de Tenerife	1,4645079601
Santiago de Compostela	1,716906855
Sevilla	1,637789855
Toledo	1,2370588531
Valencia	1,4722509824
Zaragoza	1,2805200341

Tabla 6.1 Resultado MAE estadístico por ciudades

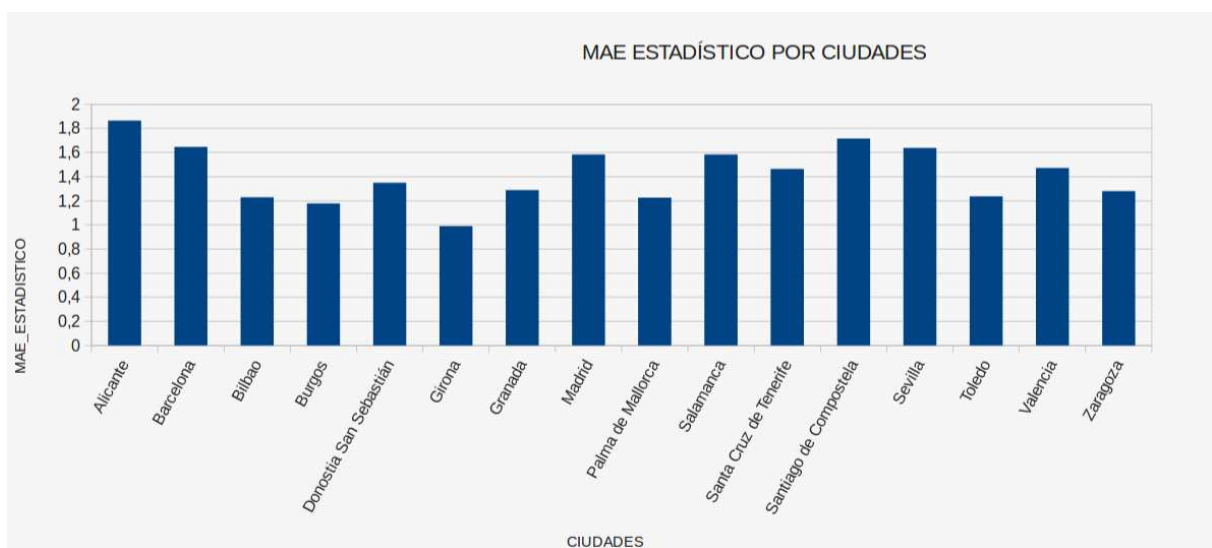


Figura 6.7 Grafica MAE estadístico por ciudades

Tabla que contiene la media del MAE por cantidad de estrellas.

Stars	Cantidad	Media
1	25	2,9181093239
1,5	0	0
2	36	2,068890187
2,5	0	0
3	22	1,1791741737
3,5	12	0,764729674
4	23	0,4334469026
4,5	15	0,6588907316
5	26	0,9606594763

Tabla 6.2 Resultado MAE estadístico según categoría (estrellas)

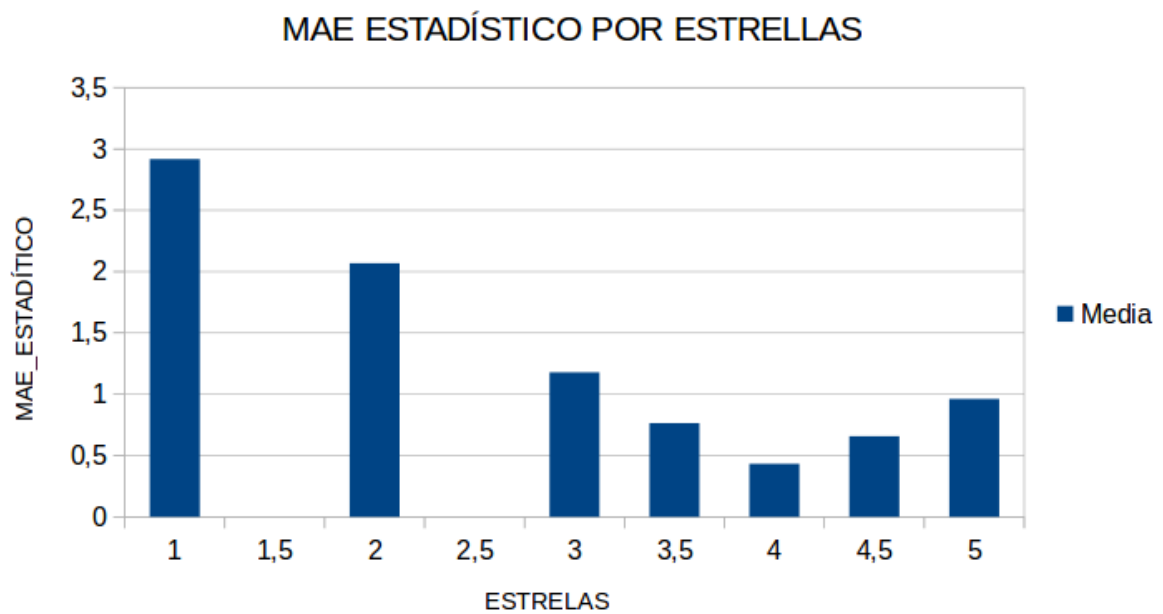


Figura 6.8 Gráfica MAE estadístico según categoría (estrellas)

Según la gráfica podemos observar que nuestros resultados se aproximan mucho a los obtenidos por los TripAdvisor y Trivago, para los hoteles de media y alta categoría, de esto se debe a que logramos encontrar más información (comentarios, palabras claves).

6.2.2 Estudio del análisis sentimental

Tabla que contiene la media del MAE por ciudades³⁷.

Ciudades	MAE Sentimental
Alicante	1,4427357359
Barcelona	1,971378624
Bilbao	1,4400237455
Burgos	1,8175412955
Donostia San Sebastián	1,5583277194
Girona	1,4971786967
Granada	1,2582433608
Madrid	1,7692415184
Palma de Mallorca	1,5872069969
Salamanca	1,2399523907
Santa Cruz de Tenerife	1,6410327646
Santiago de Compostela	1,5581456105

³⁷ <http://mlh.fdi.ucm.es/SRH/testSentimental.pdf>

Sevilla	1,3109732561
Toledo	1,2655407229
Valencia	1,3955411114
Zaragoza	1,1470317882

Tabla 6.3 Resultado MAE sentimental por ciudades

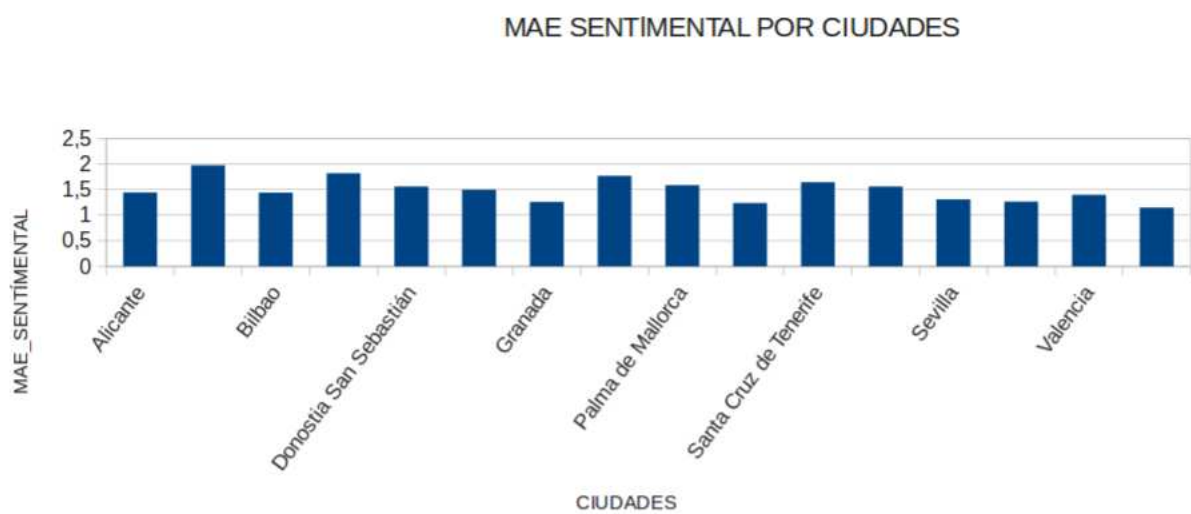


Figura 6.9 Gráfica MAE sentimental por ciudades

Tabla que contiene la media del MAE por cantidad de estrellas.

STARS	CANTIDAD	MAE Sentimental
1	25	2,529809833
1,5	0	0
2	36	1,6831519385
2,5	0	0

3	22	0,5704327199
3,5	12	0,7233888597
4	23	0,8733776149
4,5	15	1,5427720607
5	26	1,9351320284

Tabla 6.4 Resultado MAE sentimental según categoría (estrellas)

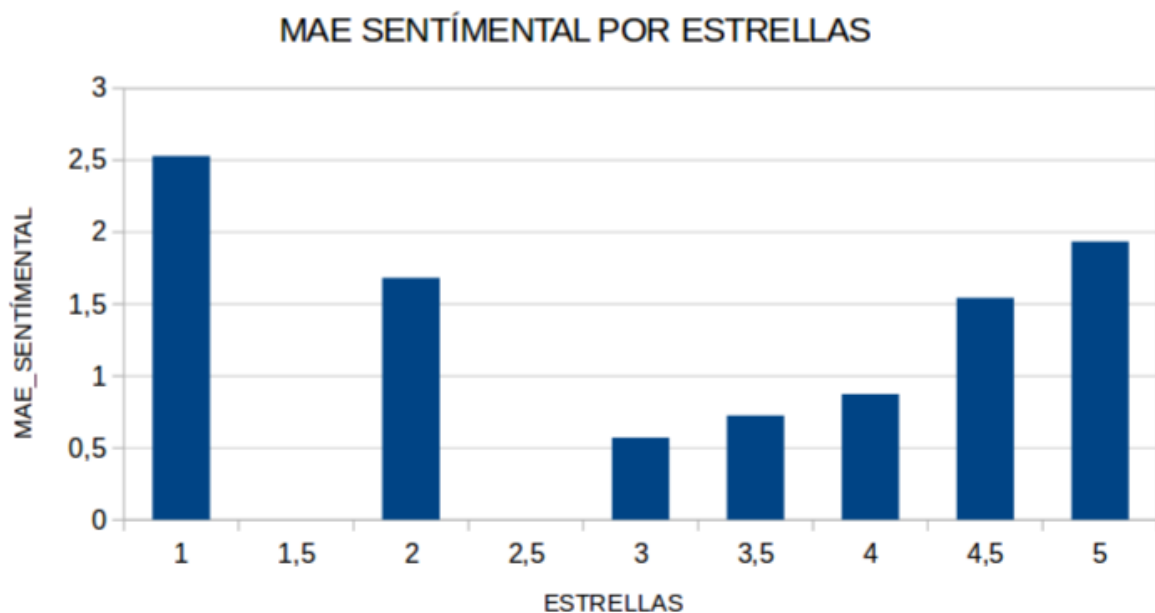


Figura 6.10 Gráfica MAE sentimental según categoría (estrellas)

6.2.3 Estadístico vs Sentimental

Tabla que contiene los resultados³⁸ medios por ciudad.

³⁸ <http://mlh.fdi.ucm.es/SRH/testEstadisticoVSsentimental.pdf>

Según ciudades

CIUDADES	MAE_ESTADÍSTICO	MAE_SENTIMENTAL
Alicante	1,8651404897	1,4427357359
Barcelona	1,6461272897	1,971378624
Bilbao	1,2290894886	1,4400237455
Burgos	1,1776858452	1,8175412955
Donostia San Sebastián	1,3498304307	1,5583277194
Girona	0,9898301973	1,4971786967
Granada	1,2889542969	1,2582433608
Madrid	1,5837431425	1,7692415184
Palma de Mallorca	1,2258486963	1,5872069969
Salamanca	1,5843694312	1,2399523907
Santa Cruz de Tenerife	1,4645079601	1,6410327646
Santiago de Compostela	1,716906855	1,5581456105
Sevilla	1,637789855	1,3109732561
Toledo	1,2370588531	1,2655407229
Valencia	1,4722509824	1,3955411114
Zaragoza	1,2805200341	1,1470317882

Tabla 6.5 MAE estadístico vs Sentimental por ciudades

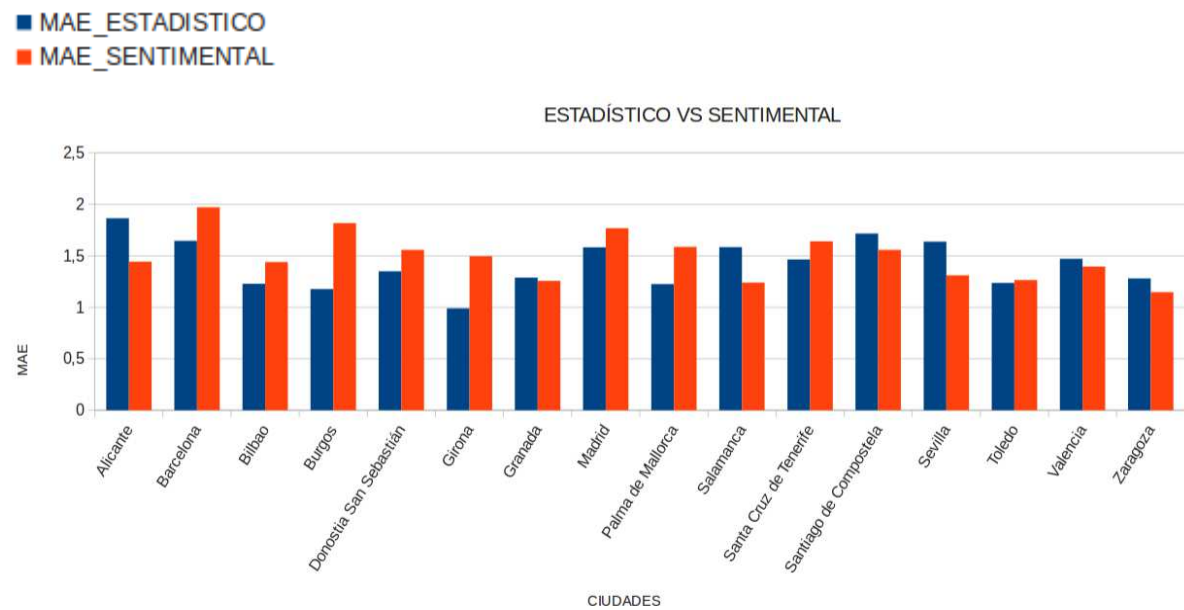


Figura 6.11 Gráfica MAE estadístico vs sentimental por ciudades

Según Estrellas.

ESTRELLAS	MAE_ESTADÍSTICO	MAE_SENTIMENTAL
1	2,9181093239	2,529809833
1,5	0	0
2	2,068890187	1,6831519385
2,5	0	0
3	1,1791741737	0,5704327199
3,5	0,764729674	0,7233888597
4	0,4334469026	0,8733776149
4,5	0,6588907316	1,5427720607

5	0,9606594763	1,9351320284
---	--------------	--------------

Tabla 6.6 MAE estadístico vs Sentimental según categoría (estrellas)

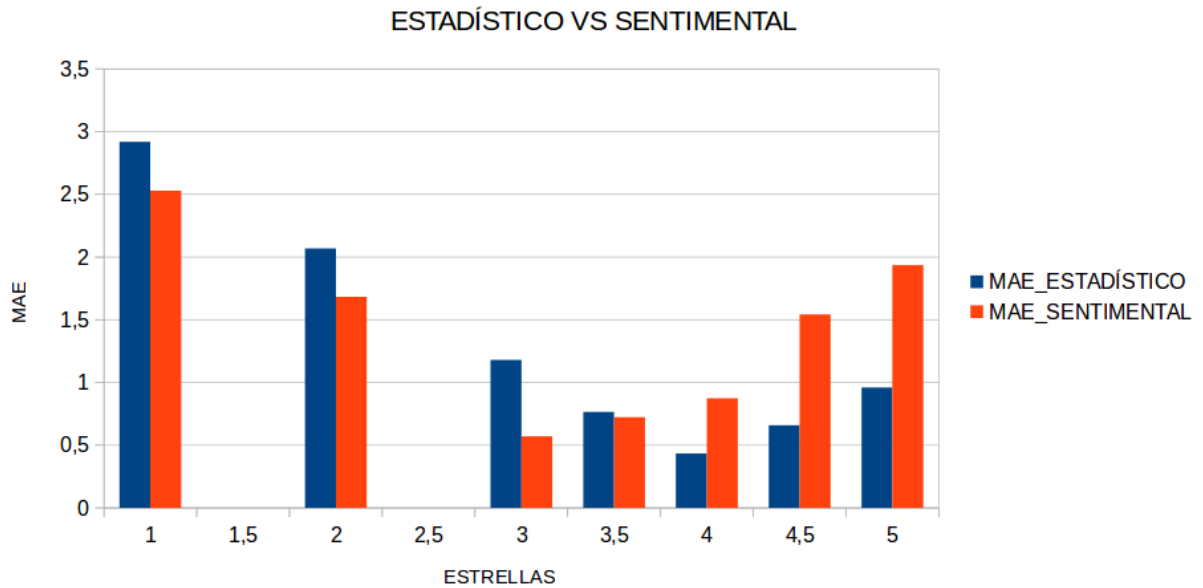


Figura 6.12 Gráfica MAE estadístico vs Sentimental según categoría (estrellas)

Podemos observar que los resultados obtenidos del Análisis Sentimental son más precisos que los obtenidos por el Análisis Estadístico a la hora de evaluar hoteles de entre 1 y 3,5 estrellas de puntuación, ocurriendo lo contrario para el resto de hoteles, esto se debe que Google encuentra buenos resultados para los hoteles de 4 a 5 estrellas.

7 Conclusiones y Trabajo Futuro

7.1 Conclusiones

Una vez acabado el proyecto, y haber pasado por varios dolores de cabeza sacamos como conclusión el hecho de haber podido aprender el uso de muchas herramientas y tecnologías para el desarrollo del software, haber adquirido mucho conocimiento y habernos enfrentado a nuevos problemas a los cuales hemos buscado una solución, solución que en un principio no fue nada fácil.

En un inicio no supimos en qué mundo nos habíamos metido, pero a medida que íbamos investigando nos surgieron diversas preguntas por responder, preguntas como las siguientes:

¿Para qué sirve lo que estamos haciendo?

¿A quiénes les interesaría nuestro proyecto?

¿Existen sistemas desarrollados con la misma funcionalidad ?

En el caso de que existieran, ¿En que se diferenciaría nuestro sistema de los existentes?

A medida que avanzábamos en el desarrollo de nuestro proyecto las íbamos respondiendo a todas ellas.

Solo al terminar este proyecto nos hemos dado cuenta de algo muy interesante, que es el impacto socio-económico que deja nuestro SRH como otros recomendadores de servicios o productos pues lo que muestra en un plano paralelo es la popularidad y el prestigio de un sitio de alojamiento, al cual mucha gente irá o dejará de ir por las valoraciones que se obtenga del hotel, entran aquí muchos campos a los que ya no nos dedicamos, como campañas de publicidad, marketing, intelligentbusiness, etc., todo esto a efectos de la valoración de nuestros analizadores.

De lo anterior los propietarios, administradores o gerentes de hoteles se verán obligados a mantener el status y hasta retroalimentar su modelo de negocio, por comentarios negativos y bajas puntuaciones.

Saber que el trabajo con expresiones regulares, el manejo del lenguaje en castellano da lugar a una complejidad alta y que la relación entre mayor refinamiento para el analizador sentimental representa mayor trabajo con el diccionario de datos y es tarea mayormente compleja.

Un sistema recomendador dará al usuario una verdadera interpretación del producto o servicio, si el fin es un beneficio económico puede este verse comprometido a ser

independiente y mitigar reacciones de críticas en los comentarios de los usuarios o simplemente desaparecerlas, por eso creemos aconsejable que la información sea extraída de foros y sitios independientes.

7.2 Trabajo Futuro

Concretar un trabajo a futuro es un poco complicado ya que a estas alturas se nos ocurren infinitudes de cosas que podemos hacer y adaptarlas a nuestro sistema, entre las cuales se encuentran, cosas como:

- Ampliar nuestro diccionario de palabras y frases, almacenando dicha información en una base de datos noSQL³⁹, para así de esa manera tener un sistema que sea escalable.
- Almacenar enlaces a los foros y blogs en los cuales se encuentran la gran cantidad de comentarios, para así centrarse solo en webs específicas de las cuales poder extraer gran volumen y variedad de datos.
- Ampliar nuestra API para así brindar un mayor servicio para todas las personas interesadas en desarrollar sus propias aplicaciones.
- Algo que tendría sentido es saber verificar de alguna manera que los mensajes o comentarios negativos son precisamente de usuarios que han tenido la experiencia de alojarse en un hotel y diferenciarlo de cualquier persona que se queje solo por quejarse.
- Si se ha logrado el paso anterior se puede refinar el análisis de expresiones regulares a tal punto que sepa cuándo y cuando no obedecer a la réplica de un hotel cuando se defiende de una crítica.
- Se podrían definir más perfiles de usuarios y a estos estudiar las características correspondientes y qué peso darles a la hora de calcular la valoración, esto ampliaría nuestro trabajo obteniendo más filtros para cumplir la satisfacción de varias opciones o necesidades de nuevos clientes.
- Se podría refinar el trabajo del analizador estadístico con mayor número de features, lo que representaría componer un modelado matemático más amplio y ni hablar de cómo analizar los resultados con los reales haciendo test como los que hemos presentado.

³⁹ <https://www.mongodb.org/>

Conclusions and Future Work

Conclusions

Once finished the work, and after going through several headaches, we give as a conclusion the fact that we have been able to learn the use of several tools and technologies to the software's development, we have acquired a lot of knowledge and we have faced new problems, to which we have looked for a solution. In the beginning, looking for this solution wasn't an easy thing to do.

In the beginning we didn't know where we had got ourselves into, but as we went investigating, a lot of unanswered questions came up, like:

¿To what purpose are we doing this project?

¿To whom would interest what we're doing?

¿Are there any development systems with the same functionality?

¿ If there are, on what would our project be different?

As we went on developing our project, we were answering all of these questions.

Just by finishing this project, we are aware of something really interesting, which is the social-economic impact that our SHR, like other recommendation services or products, because what is shown is a parallel between the popularity and the prestige of a place to stay, to which a lot of people will go to not go depending on the valuations that the hotel gets, and here we have a lot of fields that we do not dedicate to like advertising campaigns, marketing, intelligentbusiness, etc., all of which affects the valuation of our analyzers.

The owners, the administrators or hotel managers will see themselves obligated to keep the status and even retorted their business model, because of negative comments and low scores.

Knowing that the work with regular expressions, managing the spanish language is a work of high complexity, in which a bigger refinement to the sentimental analyzer represents a bigger amount of work with the dictionary of data, which is a more complex task.

A recommendation system will give the user a true interpretation of the product or service, if the objective is an economical benefit this one's independency could be compromised, and it could mitigate criticisms on the users' comments, or just make them disappear, and because of this, we believe that it is advisable that the information is extracted from forums and independent websites.

Future Work

To think of the possibility of a future work is a bit complicated because at this point there are an infinite number of things we can do and adapt them to our system, among which are:

- Expand our dictionary of words and phrases, storing this information in a NoSQL database, so that way to have a system that is scalable.
- Save links to forums and blogs in which there are a large number of comments, focusing only on specific sites to extract large volume and variety of data.
- Expanding our API in order to provide a better service for all people interested in developing their own applications.
- Something that would make sense is knowing somehow if the messages or negative comments are really from users who have had experienced staying at a hotel and differentiate it from anyone who complains just to complain.
- If the previous step is achieved, the analysis of regular expressions can be refined to the extent to the point of knowing when and when not to obey the replica of a hotel when defending criticism.
- Define more user profiles and to those, study the corresponding characteristics and what weight to give to the calculation of the assessment, this would expand our work getting more filters to meet the satisfaction of several options or needs of new customers.

Refine the work of the statistical analyzer with an higher number of features, which would compose a broader mathematical modeling let alone how to analyze the results, doing test as we the ones we have presented.

8 Apéndice A

8.1 Contribuciones y trabajo individual

8.1.1 Eddy Cuizaguana Cerpa

Empiezo listando todo el trabajo que hemos/he realizado a lo largo del curso para así finalmente llegar a nuestro objetivo, el desarrollo de una aplicación que recomienda hoteles, son:

- Búsqueda de Recomendadores de Hoteles.
- Búsqueda de APIS públicas relacionados con los hoteles, sus servicios y las opiniones que realiza la gente sobre los mismos.
- Investigación Google
- Investigación sobre las diversas tecnologías para el desarrollo de la aplicación.
- Reparto de trabajos
- Desarrollo del Frontend
- Desarrollo del Backend
- Realización de Test analizando los resultados en tablas y realización de gráficos.

Búsqueda de Recomendadores de Hoteles:

Para el desarrollo de nuestro proyecto hemos empezado investigando la mayoría de las aplicaciones que ofrecen servicios de recomendaciones de hoteles, así como las distintas valoraciones que puntúa a los hoteles que recomienda al público, comentarios de los clientes en los que expresan su conformidad o no del mismo, para así finalmente extraer información que podríamos analizar y usar en nuestra aplicación.

Una vez Analizados los diversos recomendadores, decidimos centrarnos en TripAdvisor y Trivago ambas pioneras en recomendar hoteles a los usuarios acorde a sus necesidades.

Teniendo ya en mano las aplicaciones a estudiar, el siguiente paso fue buscar documentación pública (API) con la cual podríamos trabajar, fue aquí donde se nos empezaba a complicar el trabajo, ya que no encontrábamos ninguna API pública de la que podríamos extraer información tales como comentarios, opiniones y descripciones de hoteles con un nivel de detalle medio, esto se debía a que las empresas sólo ofrecen su API si se realiza un pago por la misma, llegado a esta conclusión nos obligamos a ver otras posibles formas de obtener la valiosa información, y es aquí en donde nos aventuramos en maravilloso mundo de Google.

Investigación Google:

La mayoría de las personas piensa que Google es internet, este pensamiento lo tienen debido a que ante la necesidad de búsqueda de determinada información recurren al buscador de Google, sin embargo Google aparte de brindar este servicio brinda otros muchos. Es por ello que cambiamos de mentalidad pensando como la mayoría de las personas "Google lo sabe todo".

Con este cambio buscamos información sobre el funcionamiento del motor de búsqueda de Google y como con el poder realizar búsquedas avanzadas (Fustero, 2013), trabajo que me llamó mucho la atención y el cual decidí encargarme. Visitando diversos foros y blogs pude documentarme y aprender técnicas que con unos simples pasos obtenía la valiosa información que necesitábamos para el avance de nuestro proyecto, los resultados de búsqueda.

Investigación sobre las diversas tecnologías para el desarrollo de la aplicación:

Sabiendo de que tendríamos que desarrollar tanto el Frontend como el Backend de nuestra aplicación, decidir sobre qué tecnologías usar fue una labor difícil debido a que teníamos que tener en cuenta muchos factores tales como: la gran variedad de tecnologías disponibles, unas más recientes que otras, el interés por aprender, nuestro conocimiento y el tiempo que disponíamos. Entre las tecnologías y librerías que estaban a nuestra disposición fueron:

Backend:

- Jersey
- JEE
- node.js

Las dos primeras nos resultaban más fáciles debido a que ambas son plataformas Java y disponíamos de conocimientos, node.js sería un reto para nosotros ya que es una tecnología que está ganando popularidad actualmente, pero lo de programar con JavaScript en el lado del servidor no nos llamó la atención debido a que no es un lenguaje estructurado y su mantenimiento sería difícil.

El mundo del Frontend, es aquí donde mi compañero Roberto toma el mando y se responsabiliza de su desarrollo, usando diversas tecnologías, tales como:

Frontend:

- JavaScript
- JQuery
- Bootstrap
- HTML5

Reparto de trabajos

Además de los trabajos antes mencionados y asignados, quedaba por decidir quién iba a desarrollar las restantes tareas.

Debido a que mi compañero se encargaría del Frontend, el desarrollo del Backend así como el testeado de la lógica de aplicación fue mi responsabilidad y con ello el mantenimiento y la gestión del servidor otorgado por la facultad de informática.

Ya que nuestra intención era compartir todo nuestro trabajo con todo el mundo, usamos GitHub para llevar a cabo el control de versiones de nuestro código.

Desarrollo del Frontend:

Es aquí en donde mi contribución fue muy ya que disponía de poco conocimiento, pero a pesar de ello he ayudado a tomar ciertas decisiones, a investigar ciertas funcionalidades que necesitábamos y a realizar alguna que otra prueba.

Desarrollo del Backend:

Gran parte de mi trabajo se centró en el desarrollo de esta arquitectura, para ello tuve que documentarme, investigar y probar diversas funcionalidades, estas fueron:

- Control y gestión del servidor
- Desarrollo de la aplicación en Java usando Jersey.
- Realización de test

Control y gestión del servidor: Para realizar el mismo he utilizado una conexión ssh con la que podría trabajar desde casa. Realicé una configuración inicial instalando aplicaciones como:

- Java
- Tomcat
- vim

Por diversos fallos técnicos con el servidor de la facultad, tuve que montar un servidor en casa haciendo uso de la Raspberry Pi que disponía, el mismo fue de gran ayuda para poder realizar pruebas de nuestra aplicación y de esa manera no quedarnos atascados.

Desarrollo de la aplicación en Java usando Jersey: Teniendo en cuenta que necesitábamos implementar un servicio REST investigando y buscando me he topado con Jersey, la cual con un poco de paciencia y tiempo pude en un par de días aprender cómo sacarle un buen uso y conseguir lo que buscábamos.

Hice uso de diversas librerías tales como:

- jersey-server-1.0.3.jar
- gson-2.2.1.jar
- jersey-core-1.0.3.jar
- jsoup-1.8.2.jar
- jsr311-api-1.0.jar
- poi-3.11-beta2.jar

Las cuales fueron de gran aporte para así obtener el JSON que sería enviado al Frontend.

Para que todo tuviera consistencia faltaba por construir nuestro modelo, trabajo el cual realizaría nuestra lógica de aplicación, que constaba de lo siguiente, dado unas valoraciones de un hotel y los resultados encontrados en foros y blogs, relacionarlos y sacar una conclusión

que podría responder a la siguiente preguntas ¿Qué tan bueno es el hotel?, ¿Que de bueno son nuestras puntuaciones con respecto a las de TripAdvisor y Trivago?, para responder a las mismas he realizado el cálculo del M.A.E (error medio cuadrático) . Explicado en pocas líneas parece sencillo, pero no lo fue, ya que es aquí donde se centraba el gran funcionamiento de nuestro proyecto, y es por ello que no se logró terminar semanas antes de su entrega.

Realización de test: Para verificar la certeza de nuestro algoritmo he realizado un test en el que seleccionando una serie de hoteles, concretamente 10 por ciudad y dos por cantidad de puntuación los he almacenado en un fichero para su posterior análisis. Dicho test se complicaba ya que Google limitaba en el número de peticiones, fue en esos momento donde empecé a investigar cómo evadir dicha restricción, por el camino me topé con Orchid, librería implementada en Java, la cual me permitía realizar una conexión a TOR como cliente, pero viendo que el avance del mismo era escaso, e investigando un poco más, logré dar con las conexiones a través de un proxy, el cual con un poco más de tiempo para realizar las peticiones a Google me solucionó por completo el problema y así poder obtener los resultados para el test.

Para finalizar, con la ayuda de la librería Poi he pasado los resultados a hojas Excel para facilitar su lectura y comprensión con la ayuda de gráficas de barras.

8.1.2 Roberto Yáñez Torres

En cuanto a mi trabajo personal y mi contribución con nuestro SRH, lo detallo a continuación, sin dejar de lado la gran labor que como equipo hemos tenido con mi compañero, el intercambio de ideas, las reuniones y las largas charlas hasta altas horas por Skype han dado buen resultado con llevar a un buen punto final esta aplicación.

Como en principio no sabíamos cuál sería el modo de presentar nuestra aplicación (desarrollo del Frontend) del que me ocuparía en totalidad más adelante, las siguientes tareas han sido comunes y otras divididas para agilizar avances de nuestro proyecto, que gracias al ritmo que llevamos de trabajo para cumplir una presentación semanal a nuestro tutor Dr. Juan Antonio nos mantenían muy ocupados y siempre en acción.

Mi contribución:

- Búsqueda de información de hoteles, ver ¿qué comenta la gente?, revisión de foros, blogs, redes sociales, etc._(Magicien, 2013).
- Búsqueda de APIs públicas para servicios de obtención de contenido web como es RSS, REST o Atom y ScreenScraping_(ProgrammableWeb, 2015)_ (tuderechoasaber, 2015) (GoogleDevelopers, 2015).
- Desarrollo del primer programa en Java de obtención de contenido web, dada una url, la idea era ir probando y ver qué utilidad nos podría dar y que tipo de librerías del lado del Backend podríamos usar: java.net.URL, java.net.URLConnection;

- Búsqueda de recomendadores y buscadores de hoteles, entre varios finalmente nos quedamos con los más robustos, los cuales nos daban mayor información e ideas para nuestro desarrollo.
- Búsqueda, análisis y pruebas de APIS de Google _(Google, 2015) _(GooglePlaceAPI, 2015), ya con ideas cada vez más claras fuimos investigando sobre las bondades de Google y sus APIs, que nos ayudaría en producción.
- Creación de cuenta Google para nuestro proyecto, con ello manejar las APIs investigadas generar un Key desde la consola de Google, necesario para ciertas pruebas.
- Primeros encuentros con Servlets, que descartamos luego, porque no queríamos tener el código de desarrollo web, mezclado con el del servidor y otras desventajas que nos suponía, como la curva de aprendizaje muy alta.
- Trabajo en Java con la librería webbridge con la cual ya fuimos teniendo en cuenta que el Backend podría funcionar totalmente en Java y cuya conexión con el Frontend sería Json bajo el trabajo de Jersey.
- Desarrollo e implementación de la estimación estadística de acuerdo al Teorema de Bayes, esta implementación más tarde fue heredada a mi compañero quien se encargó del desarrollo en el servidor.

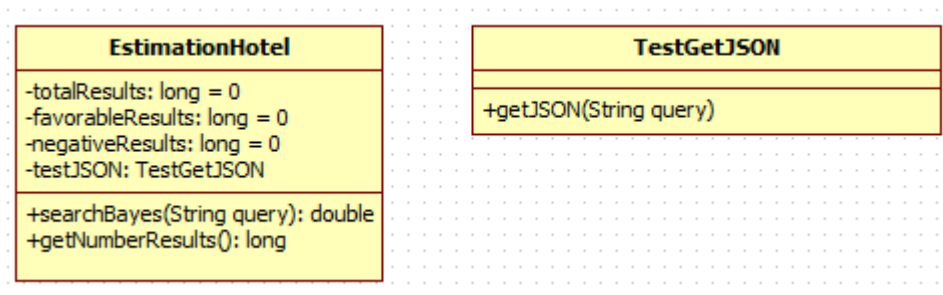


Figura 8.1 Teorema de Bayes

- Investigación de Jersey, vista de ejemplos “HelloWorld” _(Mkyong, 2012).
- Búsqueda, análisis e implementación de pequeños programas de prueba en Java con analizadores léxicos, recursos para adentrarnos a la implementación con las expresiones regulares que luego fueron implementadas para el funcionamiento del Analizador Sentimental, librerías usadas JFlex.
- Trabajo de investigación del uso frecuente de qué palabras son las más usadas en los blogs y foros por los clientes de hoteles, categorización y clasificación de keywords y expresiones regulares positivas y negativas, trabajo un poco a mano y también uso de herramientas como Google Trends, para contrastar si las palabras seleccionadas eran las correctas en la tendencia en cuanto a comentarios para hoteles.
- Análisis para la implementación de los algoritmos de valoración y recomendación Estadístico y Sentimental (reuniones de horas en laboratorios y biblioteca con mi compañero).
- Colaboración del diccionario de palabras para el Analizador Sentimental, manejo de expresiones regulares, usando keywords positivas y negativas.

- Creación web de un buscador casero de Google, el objetivo era aprender el uso del API de Google Search y saber los componentes web y el manejo interno con JavaScript y HTML.

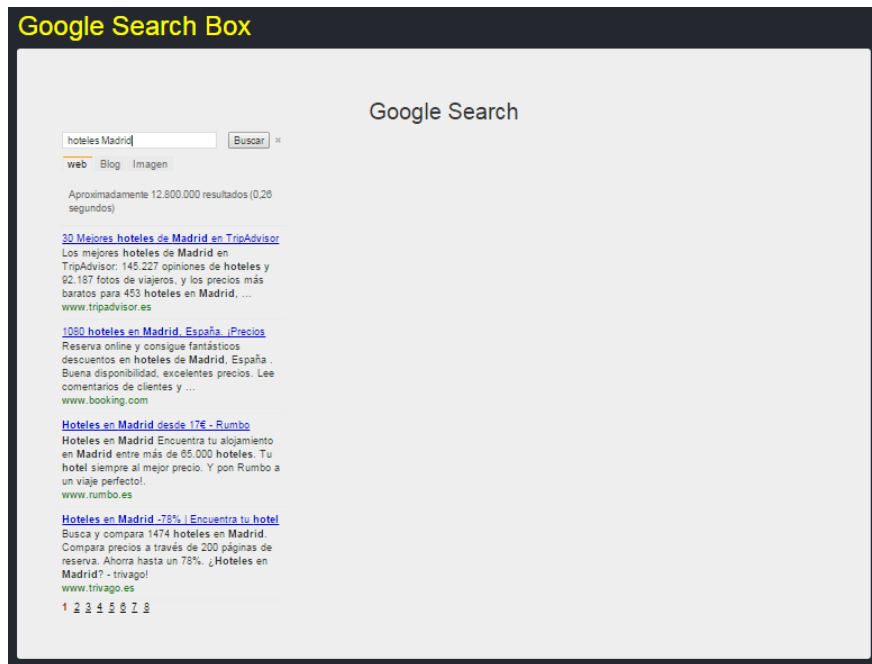


Figura 8.2 Google Search Box

- Pruebas con JavaScript y con el API de Google Maps para controles y eventos con el mapa de Google.
- Creación web para buscar sitios en el mapa de Google, objetivo familiarizarse con el API de Google Place, y sacarle el mayor potencial para las necesidades de nuestros usuarios, en este caso no solo podía escoger hoteles, sino tiendas, restaurantes, museos, etc.

Google Map Place Searches

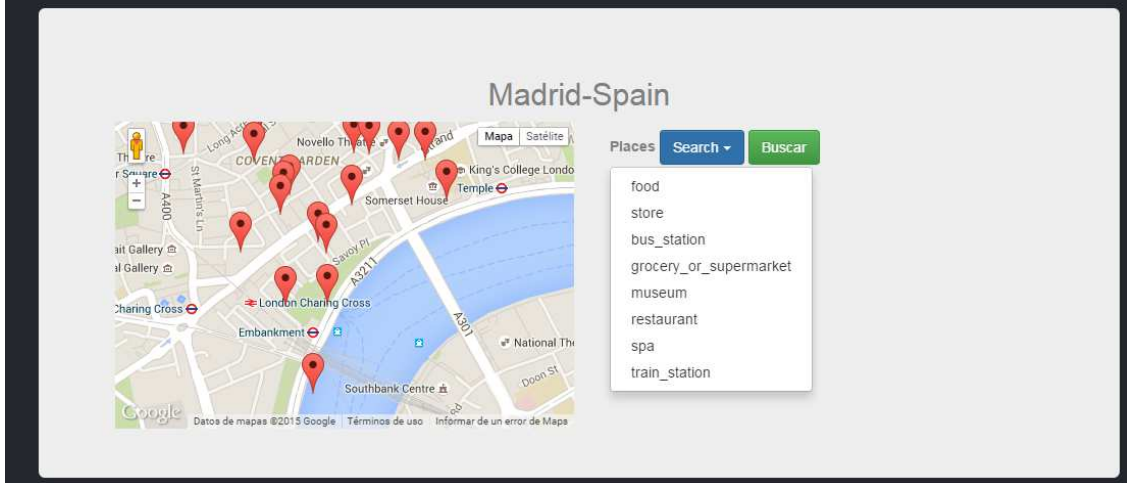


Figura 8.3 Google Search Places

- Colaboración para los test y comparaciones del MAE, extrayendo a mano una muestra de considerable de hoteles, con sus puntuaciones.
- Aprovechando las capacidades individuales y haciéndonos fuertes en el divide y vencerás, mientras mi compañero va haciendo un gran trabajo con el Backend el servidor y solucionando problemas de detección de bots por Google con herramientas como TOR, por mi parte me dediqué a la tarea de hacer lucir con criterio todo lo que esconde una página web. Para ello he seguido los siguientes pasos:
 - Búsqueda de tecnologías webs a utilizar, resultando las mejores candidatas JavaScript, JQuery, HTML, CSS, Bootstrap, Material design, y trabajos con APIs de Google Maps, Google Places, de geo localización, entre otras.
 - Búsqueda de templates de Bootstrap y elección de uno de ellos sobre el cual empezaría a realizar una personalización y la implementación de los componentes para la interacción del usuario con nuestro SRH.
 - Diseño en papel de componentes y distribución del contenido a mostrar, interacciones, animaciones y feedbacks.
 - Primer boceto de la página web, problemas con el mapa de Google con el responsivedesign, solucionado más tarde con una nueva programación.
 - Segundo y final boceto con cambios de posición de contenedores, diseño de botones, elección de imágenes del Header y el Footer.
 - Uso de Materialsdesign, y primeros problemas con las hojas de estilo de cada tecnología, chocaban archivos y el diseño fue una pesadilla, pero el resultado ha valido la pena, personalmente la mezcla del diseño plano que ofrece Bootstrap más la profundidad, manejo de sombras y altos relieves de Materialsdesign han podido convivir en esta aplicación con paciencia en el diseño y siendo detallista.

- Una vez puesto todo sobre la web, faltaba la funcionalidad, y entramos en el campo de estudio de Json y el parseo del mismo que viene desde el Backend y para ello nada mejor y más fácil que el manejo de JQuery.
- Lo siguiente era mostrar las valoraciones que como es de imaginarse deberá llamar la atención del usuario una vez salir, para ello funciones con JavaScript nos ayudó a mostrar estrellas que muestran el resultado dinámicamente con el valor que facilita el Json enviado desde el Backend, esto funciona tanto para el Análisis Estadístico, como para el Análisis Sentimental.
- Trabajo con el API de Google Maps para eventos en el mapa como mostrar posición actual por: interacción directa con el mapa, con elección de una ciudad desde un combo box externo, y también con la creación de un botón propio, añadido al mapa como nuevo componente para la geo localización usando el GPS. Otro de los eventos fue mostrar marcadores personalizados de posición, de hoteles y de ciudades, añadiendo en este texto informativo nombres o links en el caso del escudo de la ciudad que se muestra y al darle click da información complementaria con datos de la ciudad en Wikipedia.
- Manejar eventos en toda la página como autocompletación del nombre del hotel seleccionado desde el mapa al formulario para los futuros análisis de evaluación estadístico o sentimental.
- Trabajar con animación con CSS para los botones y con JavaScript para los feedbacks de espera.
- Y como cereza del pastel he diseñado un favicon que da ícono gráfico en la pestaña de navegación del browser, muy fácil usando servicios online de diseño_(GmbH, 2014).

9 Bibliografía

- ABC. (22 de 01 de 2014). *ABC*. Recuperado el 2015, de ABC:
<http://www.abc.es/economia/20140121/abci-turistas-record-201401211049.html>
- Alfredo. (10 de 04 de 2013). *Críticas de Hoteles*. Recuperado el 2015, de Críticas de Hoteles:
<http://criticasdehoteles.blogspot.com.es/>
- Barbero, F. L. (15 de 11 de 2012). *Ubuntu León*. Recuperado el 2015, de Ubuntu León:
<http://www.ubuntuleon.com/2012/11/como-navegar-de-forma-anonima-en-ubuntu.html>
- Corporation, O. (24 de 03 de 2015). *Jersey*. Recuperado el 2015, de Jersey:
<https://jersey.java.net/>
- Fuentes, J. J. (2010). *Sistema Recomendador Turístico*. Recuperado el 2015, de Sistema Recomendador Turístico:
http://aulavirtual.tecnologicocomfenalcovirtual.edu.co/aulavirtual/pluginfile.php/74872/mod_resource/content/1/Varios/Sistema_Recomendador_Turistico.pdf
- Fustero, E. M. (18 de 06 de 2013). *Comunidad IEBS*. Recuperado el 2015, de Comunidad IEBS:
<http://comunidad.iebschool.com/iebs/general/comandos-busqueda-avanzada-google/>
- Gallardo, J. C. (2012). *sinbad2*. Obtenido de sinbad2:
http://sinbad2.ujaen.es/cod/archivosPublicos/dea/TTII_JorgeCastro.pdf
- GmbH, X. (2014). *favicon.cc*. Recuperado el 2015, de favicon.cc: <http://www.favicon.cc/>
- Google. (2015). *Google Maps Engine API v1 (revision 60)*. Recuperado el 2015, de Google Maps Engine API v1 (revision 60): <https://developers.google.com/resources/api-libraries/documentation/mapsengine/v1/java/latest/>
- GoogleDevelopers. (2015). *Google Developers*. Recuperado el 2015, de Google Developers:
https://developers.google.com/places/supported_types
- GooglePlaceAPI. (2015). *Google Place API*. Recuperado el 2015, de Google Place API:
<https://developers.google.com/places/>
- INE. (22 de 05 de 2015). *Instituto Nacional de Estadística*. Recuperado el 2015, de Instituto Nacional de Estadística:
<http://www.ine.es/jaxi/menu.do?type=pcaxis&path=%2Ft11/e162eoh&file=inebase&L=0>
- Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2011). *Recommender Systems An Introduction*. New York: Cambridge University Press.
- López, M. (01 de 12 de 2010). *Familias en ruta*. Recuperado el 2015, de Familias en ruta:
<http://familiasenruta.com/alojamiento/hoteles/hoteles-free-child-nins-prohibidos/>

- Magicien, J. (19 de 01 de 2013). *VOGUE*. Recuperado el 2015, de VOGUE:
<http://foros.vogue.es/viewtopic.php?f=6&t=198287>
- Mkyong. (29 de 08 de 2012). *Mkyong.com*. Recuperado el 2015, de Mkyong.com:
<http://www.mkyong.com/webservices/jax-rs/jersey-hello-world-example/>
- ProgrammableWeb. (2015). *ProgrammableWeb*. Recuperado el 2015, de ProgrammableWeb:
<http://www.programmableweb.com/>
- Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to Recommender Systems*. Obtenido de Introduction to Recommender Systems: <http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf>
- Standard, E.-4. T. (2015). *Introducción a JSON*. Recuperado el 2015, de Introducción a JSON:
<http://json.org/json-es.html>
- TIOBE. (2015). *TIOBE Software*. Recuperado el 2015, de TIOBE Software:
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- tuderechoasaber. (2015). *tuderechoasaber*. Recuperado el 2015, de tuderechoasaber:
<http://www.tuderechoasaber.es/es/help/api>
- Wikipedia. (2015). *Wikipedia*. Recuperado el 2015, de Wikipedia:
http://es.wikipedia.org/wiki/Sistema_de_recomendaci%C3%B3n
- Wikipedia. (19 de 05 de 2015). *Wikipedia*. Recuperado el 2015, de Wikipedia:
<http://es.wikipedia.org/wiki/HTML5>
- Wikipedia. (01 de 05 de 2015). *Wikipedia*. Recuperado el 2015, de Wikipedia:
http://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada
- Wikipedia. (29 de 05 de 2015). *Wikipedia*. Recuperado el 2015, de Wikipedia:
<http://es.wikipedia.org/wiki/JavaScript>
- Wikipedia. (6 de 06 de 2015). *Wikipedia*. Recuperado el 2015, de Wikipedia:
<http://es.wikipedia.org/wiki/JQuery>
- Wikipedia. (27 de 05 de 2015). *Wikipedia*. Recuperado el 2015, de Wikipedia:
http://es.wikipedia.org/wiki/Twitter_Bootstrap
- Wikipedia. (29 de 04 de 2015). *Wikipedia*. Recuperado el 2015, de Wikipedia:
http://es.wikipedia.org/wiki/Material_design